



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV VÝROBNÍCH STROJŮ, SYSTÉMŮ A ROBOTIKY

INSTITUTE OF PRODUCTION MACHINES, SYSTEMS AND ROBOTICS

**NÁVRH A IMPLEMENTACE SW PRO ŘÍZENÍ ROBOTICKÉ
BUŇKY**

DESIGN AND IMPLEMENTATION OF THE CONTROL FOR THE ROBOTIC CELL

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Patrik Mbontar

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Tomáš Kubela

BRNO 2021

Zadaní diplomové práce

Ústav: Ústav výrobních strojů, systémů a robotiky
Student: **Bc. Patrik Mbontar**
Studijní program: Strojní inženýrství
Studijní obor: Výrobní stroje, systémy a roboty
Vedoucí práce: **Ing. Tomáš Kubela**
Akademický rok: 2020/21

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

Návrh a implementace SW pro řízení robotické buňky

Stručná charakteristika problematiky úkolu:

Cílem je navrhnout a implementovat SW včetně uživatelského rozhraní pro řízení robotické buňky se dvěma roboty KUKA a jedním pásovým dopravníkem. Princip činnosti buňky bude spočívat ve vykonávání sady manipulačních úloh řízených nadřazeným PLC systémem. Návrh a implementace SW bude probíhat na SW platformě Twincat PLC systémů Beckhoff a robotickém systému KUKA.

Cíle diplomové práce:

Systémový rozbor zadané problematiky.
Navrhnout SW pro řízení zadané robotické buňky.
Navrhnout uživatelské rozhraní pro řízení buňky.
Implementovat navržený SW.
Ověřit funkčnost SW na reálné robotické buňce.

Seznam doporučené literatury:

NOF, Y. Shimon, ed. Springer Handbook of Automation. Springer-Verlag Berlin Heidelberg, 2009. 1812 s. ISBN 978-3-540-78830-0.

SICILIANO, Bruno, KHATIB, Oussama, ed. Springer Handbook of Robotics. Springer-Verlag New York, Inc., 2008. 1611 s. ISBN 978-3-540-23957-4.

Beckhoff [online]. Beckhoff Information System: 2019 [cit.: 16.9.2019]. Dostupné z: <http://infosys.beckhoff.com/>

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2020/21

V Brně, dne

L. S.

doc. Ing. Petr Blecha, Ph.D.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

ABSTRAKT

Diplomová práca pojednáva o návrhu softwaru pre ovládanie robotickej bunky, ktorá pozostáva z dvoch robotov KUKA a pásového dopravníku. Roboty budú vykonávať robotické operácie podobné operáciám v priemyselnej praxi. Na začiatku práce bude stav súčasného poznania a rozbor riešenej problematiky. Kľúčovou časťou bude návrh softwaru pre riadenie robotickej bunky v prostredí TwinCAT od firmy Beckhoff. Roboty a dopravník budú riadene nadradeným PLC, ktoré bude povelované z navrhnutého softwaru pomocou HMI. Cieľom je úspešne implementovať navrhnutý software na riadenie robotickej bunky a monitorovanie pomocou unikátnej vizualizácie, ktorá bude navrhnutá presne pre danú robotickú bunku a jej prisluhujúcim manipulačným operáciám.

ABSTRACT

This master's thesis deals with the design of software for the control of a robotic cell, which consists of two KUKA robots and a belt conveyor. The robots will perform robotic operations similar to those in real industrial practice. At the beginning of the work will be the state of current knowledge. A key part will be the design of software for robotic cell control in the TwinCAT environment from Beckhoff. The robots and the conveyor will be controlled by a superior PLC, which will be commanded from the designed software using the HMI. The goal is to successfully implement the proposed software for robotic cell control and monitoring using a unique visualization, which will be designed specifically for the robotic cell and its associated manipulation operations..

KĽÚČOVÉ SLOVÁ

Software, PLC, riadenie, signál, robot, TwinCAT, bezpečnosť, program, stavový automat

KEYWORDS

Software, PLC, control, signal, robot, TwinCAT, safety, program, state machine

BIBLIOGRAFICKÁ CITÁCIA

MBONTAR, Patrik. *Návrh a implementace SW pro řízení robotické buňky* [online]. Brno, 2021 [cit. 2021-05-18]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/134886>. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav výrobních strojů, systémů a robotiky. Vedoucí práce Tomáš Kubela.

POĎAKOVANIE

Chcel by som veľmi poďakovať vedúcemu tejto diplomovej práce Ing. Tomášovi Kubelovi za ochotu, pomoc a vzácne rady či poznatky, ktoré mi pomohli dosiahnuť požadovaného cieľu. Taktiež ďakujem моjím najbližším za ich neustálu podporu.

ČESTNÉ PREHLÁSENIE

Prehlasujem, že táto práca je mojím pôvodným dielom, spracoval som ju samostatne pod vedením Ing. Tomášom Kubelom a s použitím literatúry uvedenej v zozname.

V Brne dňa 20.5.2021

.....

Mbontar Patrik

OBSAH

1	ÚVOD.....	15
2	MOTIVÁCIA.....	17
3	PREHLAD SÚČASNÉHO STAVU POZNANIA A ROZBOR RIEŠENEJ PROBLEMATIKY.....	19
3.1	Automatizácia – Robotizácia	19
3.2	Pracovisko.....	19
3.3	Dopravník	19
3.4	Zámok Euchner.....	20
3.5	Priemyselne roboty	21
3.6	Roboty KUKA KR5 ARC a KR16-2.....	22
3.7	Riadenie robotov	23
3.7.1	Siemens TIA Portal	23
3.7.2	Mitsubishi GX Works2.....	23
3.7.3	Schneider UNITY Pro / EcoStructure Control Expert	24
3.8	Beckhoff TwinCAT3	25
3.8.1	Programové prostredie v TwinCAT	26
3.8.2	Vytvorenie PLC projektu.....	27
3.8.3	Deklarácia premenných	27
3.9	Software na vytváranie užívateľského rozhrania.....	27
4	VYTVORENIE SW PRE RIADENIE ROBOTICKEJ BUNKY	29
4.1	TwinCAT	29
4.1.1	Programovacie jazyky v TwinCAT	29
4.1.2	Základný stavový automat PLC	30
4.1.3	Podporné funkčné bloky	31
4.1.4	FB_ROBOT.....	32
4.1.5	FB_CONVEYOR	33
4.1.6	FB_RECOVERY	34
4.1.7	FB_RESET_MAN.....	35
4.1.8	Program Door	35
4.1.9	Program Safety	36
4.1.10	Program MAIN.....	37
4.2	WorkVisual.....	41
4.2.1	Externá automatika	41
4.2.2	Program CELL.SRC	42
4.2.3	Vstupné a výstupné signály robotov a PLC.....	43
4.2.4	Postupnosť signálov.....	45
4.2.5	Program .dat.....	46
4.2.6	Programy .src	47
5	VYTVORENIE HMI NA RIADENIE ROBOTICKEJ BUNKY.....	47
5.1	Vytvorenie vizualizácie v programe TwinCAT.....	47
5.1.1	Hlavné okno.....	49
5.1.2	Okno robot.....	50
5.1.3	Okno Safety	50
5.1.4	Okno conveyor	51
5.1.5	Dialógové okno login	51

6	IMPLEMENTÁCIA NAVRHNUTÉHO SW	53
6.1	Topológia robotickej bunky	53
6.2	Mapovanie signálov.....	55
6.3	Nastavenie TwinCATu	55
6.4	Monitorovanie manipulácií	57
7	ZHODNOTENIE A DISKUSIA	59
8	ZÁVER.....	61
9	ZOZNAM POUŽITÝCH ZDROJOV	63
10	ZOZNAM SKRATIEK, SYMBOLOV, OBRÁZKOV A TABULIEK.....	65
10.1	Zoznam tabuliek	65
10.2	Zoznam obrázkov	65
11	ZOZNAM PRÍLOH.....	69

1 ÚVOD

V dnešnom svete je robotizácia neoddeliteľnou súčasťou drvicej väčšiny veľkých výrobných podnikov a nabera rýchlo na strmom vzostupe. Za posledných 6 rokov vzrástla výroba a predaj priemyselných robotov až o 114%. Tento trend nám prináša možnosť rýchlejšej, presnejšej, či kvalitnejšej práce, ktorú doteraz vykonávali ľudia, či jednoduché stroje ovládané ľudskou rukou. Keďže pandémie svet zužovala a naďalej zužuje pomerne dlhý čas, ukazuje sa ako veľmi je robotizácia potrebná pre plynulý chod výrobných procesov a tak aj chod celého sveta [1].

Všetko odštartoval prvý parný stroj, nasledujúc 2. priemyselnou revolúciou. Pojem automatizácia bez zásahu človekom sa začal používať v spojitosti priemyslom 3.0, kde sa koncom 70.tych rokov vyrobilo historicky prvé PLC – programovateľný logický automat, ktorý umožnil prácu procesov v cykloch.

Momentálne prežívame revolúciu 4.0, ktorá je spojená s internetovým prepojením výrobných procesov, robotov či ľudí. Zasahuje do všetkých oblastí v podobe smart domov, smart škôl, smartfónov, smart fabriek/podnikov až po celé smart mestá. Vo veľkom sa začali simulovať všemožné deje, procesy hlavne kvôli zníženiu nákladov na výrobu, predchádzaniu problémov a lepšiemu pochopeniu vďaka názornosti. Preto je dnes potrebné aby robotické pracoviská mali aj užívateľsky prívetivú komunikáciu so zákazníkmi.

Tieto robotické pracoviská sa využívajú v podnikoch, kde je kladený vysoký nárok na výrobu, hlavne na zvyšovanie produktivity a kvality. Robotické bunky sú tvorené robotmi ktoré vykonávajú činnosti napr. manipulácie s predmetmi a paletizácie, procesné napr. nanášanie lepidla, zváranie či povrchové úpravy ako povrchové kalenie a obrábanie, diagnostiky a merania predmetov pomocou sond, či montáže dielov.

Cieľom tejto diplomovej práce bude navrhnúť a implementovať vhodný software a užívateľské rozhranie pre ovládanie robotickej bunky pomocou nadradeného PLC. Práca bude obsahovať priblíženie do problematiky prostredníctvom rešerše aktuálnych poznatkov o priemyselných robotoch a ich príslušenstve, o používanom software na programovanie PLC a robotov až po druhy jednotlivých operácií, ktoré je bunka schopná vykonávať. Hlavnou súčasťou je návrh a implementácia programu pre povelovanie bunky, ktoré bude vykonávať obsluha v reálnom čase pomocou taktiež navrhnutého, vytvoreného a implementovaného užívateľského rozhrania.

2 MOTIVÁCIA

S rozvíjajúcim sa priemyslom 4.0 je potrebné riešiť jednoduché, ľahko pochopiteľné ovládanie aj ťažšie prevediteľných operácii ktoré vykonávajú priemyselné stroje a roboty. Tlak od konkurencie rastie každým dňom, požiadavky sú kladené vyššie a vyššie, preto sa neustále vyvíjajú nové systémy a programové riešenia a firmy bojujú o prvenstvo vo svete, čo im prinesie prestíž, a s ňou aj vyššie zisky.

Táto práca by mala obsiahnuť široké spektrum poznatkov nadobudnutých za štúdium z robotiky, automatizácie, programovania PLC ale aj poznatkov získaných práve pri vypracovaní tejto práce. Po dokončení práce by jej autor mal byť schopný navrhnúť nové riešenia, vyhodnotiť najlepšie z nich a následne tieto riešenia implementovať do reálnych podmienok vo výrobnej bunke a celú bunku priviesť do stavu pripravenosti na výrobu. Okrem vyššie spomínaného musí vedieť aj logicky uvažovať, lokalizovať nedostatky a navrhnúť vhodnejšie alternatívy cieľového produktu

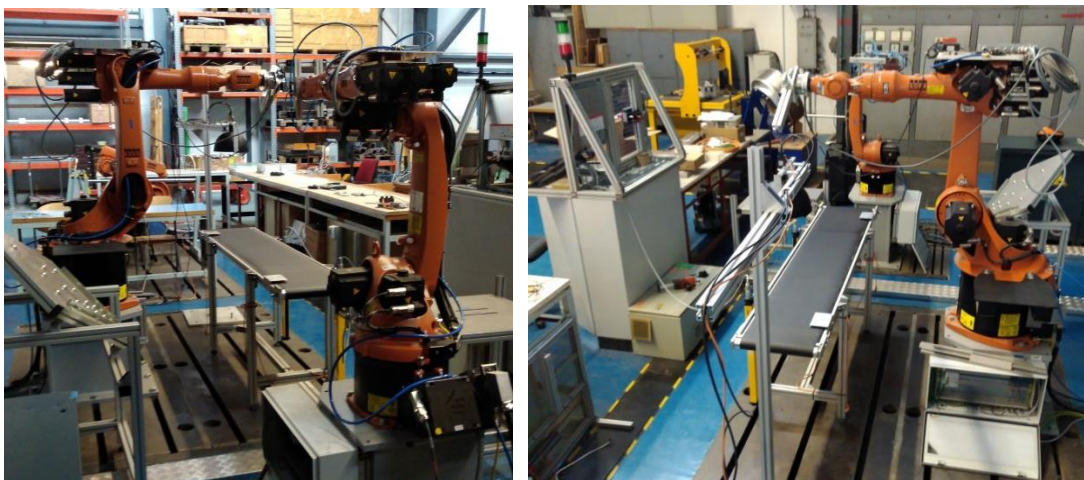
3 PREHĽAD SÚČASNÉHO STAVU POZNANIA A ROZBOR RIEŠENEJ PROBLEMATIKY

3.1 Automatizácia – Robotizácia

Robotizácia v posledných rokoch napreduje stále rýchlejším tempom a aj preto je skrátka nemožné poznať všetko v tejto oblasti. Do lepšieho obrazu tejto problematiky nás dostane práve táto kapitola. Na jej začiatku budú vo všeobecnosti opísane základne princípy a rozdelenia priemyselných robotov, ale aj bližšie popísane roboty, ktoré budú používané práve v tejto diplomovej práci. Nasledovať bude spomenutie niektorých riadiacich systémov a výrobcov PLC. Uzatvorenie tejto kapitoly bude patriť výrobcovi, ktorý dodávajú HMI (Human machine interface – Rozhranie pre komunikáciu PLC s klientom na užívateľsky prívetivej úrovni) systémy.

3.2 Pracovisko

Robotická bunka bude pozostávať ako bude bližšie rozoberané neskôr z robotov KUKA KR16-2 a KUKA KR5 ARC, pásového dopravníku DP50, riadiaceho panelu a zámku. Rozmer pracoviska je 2700 x 1980 mm. Celá robotická bunka je ukázaná na obr. 1 a, b).



Obr. 1) a), b) Robotické pracovisko

3.3 Dopravník

V tejto práci bude obsiahnutý okrem robotov aj pásový dopravník DP50, obr. 2). Dopravníky sa bežne používajú na transport predmetov napríklad medzi výrobnými pracoviskami, či celými výrobnými halami. Výhodou dopravníkov je plynulejší chod operácií, nakoľko nemusia do robotického bunky ísť ľudia kvôli manipuláciám s predmetom a tým pádom môžu roboty pracovať na plný výkon bez zastavovania kvôli vstupu človeka do bunky.



Obr. 2) Pásový dopravník DP50

V tomto prípade bude použitý na účel podávania manipulovaného predmetu medzi dvoma robotmi KUKA. Dôležité rozmery a parametre dopravníku vzhľadom k podstate práce, sú uvedené v tab. 1).

Tab 1) Základné parametre dopravníku DP50

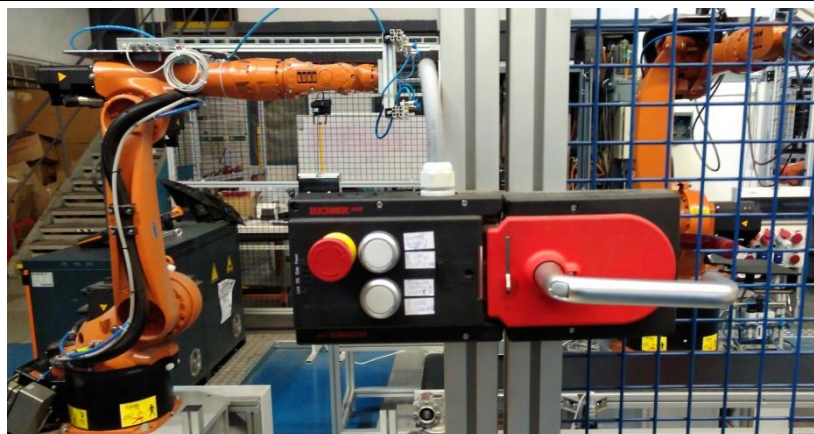
Názov veličiny	[Jednotka]	Skratka veličiny	Hodnota
Dĺžka dopravníku	[mm]	L_{dd}	2000
Šírka dopravníku	[mm]	$L_{dš}$	350
Výška dopravníku [mm]	[mm]	L_{dv}	800
Menovitý moment motoru	[Nm]	M_j	1,3
Menovité otáčky	$[\text{min}^{-1}]$	n_j	1320
Priemer hnacieho valca dopravníku	[mm]	D_{dv}	50
Prevodový pomer motoru	[-]	i	7

3.4 Zámok Euchner

Jeden z bezpečnostných (safety) prvkov bude aj zámok Euchner MGB, obr. 3). Zámok v tomto prevedení sa skladá z 2 modulov a to z blokovacieho modulu, kde sú tlačítka a kontrolky a modulu vonkajšej kľučky. Na blokovacom module sú dve tlačítka s podsvietením, ktoré sú typu spínací kontakt a majú označenie S1 a S2. Pomocou týchto tlačidiel sa dvere odomykajú a zamykajú. Taktiež je tu tlačítko núdzového zastavenia, ktoré je rozpínacieho charakteru. Toto tlačítko ma aj aretáciu formou otočenia tlačítka, čím je dosiahnuté zaistenie. Tento výrobok ma aj signalizačne výstupy :

- OD – zapnuté pri zatvorených dverách
- OT – Jazýček závery je zasunutý do vyhodnocovacieho modulu.
- OI – Diagnostika, poruchový stav

Pri zamknutí dverí sa jazýček závery zasunie do blokovacieho modulu. Pri požiadavke na otvorenie sa zase vysunie von. Vonkajšia kľučka nepôjde otvoriť, pokiaľ sú dvere zaistene avšak kľučku, ktorá sa nachádza vo vnútri bunky, je možné otvoriť aj pri zaistených dverách z bezpečnostných dôvodov – ak by bola obsluhujúca osoba nechtiac zatvorená vo vnútri bunky.



Obr. 3) Elektrický zámok bunky

3.5 Priemyselne roboty

Priemyselné roboty sa môžu rozdeľovať do kategórií podľa druhu vykonávanej operácie - lakovacie, montážne, zváracie či manipulačné roboty. Ďalšie rozdelenie je podľa generácií :

- 1.0 Generácia** - typické sú roboty prispôsobené na jednoduché operácie typu vezmi a ulož (pick and place)
- 1.5 Generácia** je význačná operáciami typu vykonaj a over (make and test). Tieto roboty majú implementované základné senzory
- 2.0 Generácia** – je vybavená už základným vnímaním a vyšším počtom senzorov na meranie viacerých fyzikálnych veličín. Tieto roboty už majú zložitejšie riadenie a sú oveľa viac automatizované
- 3.0 Generácia** robotov oplýva vysokým počtom senzorov ktoré merajú vonkajšie či vnútorné veličiny, vyhodnocujú stavy a podľa vopred napísaných inštrukcií určujú ďalší postup – autonómnosť [2]

Priemyselné roboty sa môžu deliť aj z pohľadu kinematických reťazcov, ktoré umožňujú pohyb robotu. V tomto smere poznáme 3 typy kinematických reťazcov: hybridný, paralelný a sériový.

Hybridné reťazce, tvoria otvorené a aj uzavreté kinematické reťazce, ktoré budú objasnené o pár riadkov nižšie. Do tejto kategórie patria napríklad bipody, ktoré majú vreteno polohované dvomi alebo viacerými servopodporami ako ukazuje obr. 4 a) [2].



Obr. 4) a) Hybridný robot ABB IRB 940 [4], b) Paralelný delta robot [3]

Paralelné kinematiky využívajú kĺbovo uložené podpery (3 alebo 6), ktoré sú ovládané rotačnými motormi. Niektoré kinematiky majú zase premennú dĺžku vzpier vďaka výsuvným valcom či iným lineárnym pohonom. Tieto roboty sa vyznačujú vysokou presnosťou, tuhosťou, rýchlosťou a zložitým riadením. Obr. 4 b) ukazuje robot od firmy ABB, ktorý má paralelnú kinematiku.

Sériová kinematika patrí v priemysle medzi najrozšírenejšie vo svete. Reťazec sa skladá z binárnych členov. Tieto členy sú radené za sebou – do série a vznikajú kinematické dvojice, ktoré majú v drvivej väčšine 1 stupeň voľnosti. Medzi najčastejšie používané kinematické dvojice patria translačná a hlavne rotačná [2].

3.6 Roboty KUKA KR5 ARC a KR16-2

V tejto práci budem pracovať s dvomi robotmi od firmy KUKA a to menovite s modelmi KR 5 ARC – obr. 5 a) a KR16-2 - obr. 5 b). Pôvodne mali podľa zadania tieto roboty vykonávať sadu manipulačných operácií pomocou efektoru s pneumatickým ejektorom (prísavkou) ale nakoniec jeden robot tento pneumatický uchopovač mať nebude. Oba 6-osé roboty sú postavené stacionárne na podstavci, s ktorou sa nedá hýbať. Ich dosah je približne 1,5 metra od stredu podstavu. Bližšie parametre sú vypísané v tabuľke 2).



Obr. 5) a) Kuka KR 5 ARC [4], b) Kuka KR16-2 [4]

Tab 2) Parametre robotov KR5 ARC a KR16-2

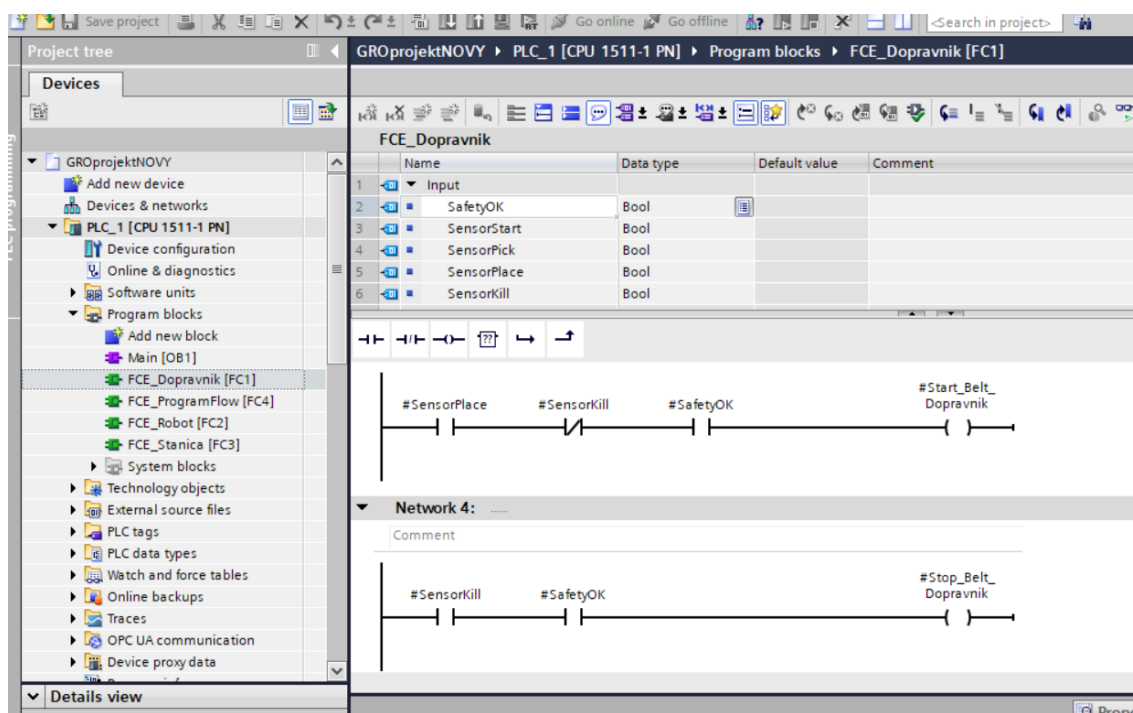
	KR 5 ARC	KR 16-2
Počet osí [-]	6	6
Menovitá nosnosť [kg]	5	16
Maximálny pracovný dosah [mm]	1530	1611
Hmotnosť robotu [kg]	127	235

3.7 Riadenie robotov

Robotické systémy sú vo veľkej miere ovládané nadriadeným PLC riadením. Výrobcovia PLC ponúkajú okrem širokej škály PLC systémov a k nim prislúchajúcim perifériám aj software, v ktorom je možné písať program pomocou programovacích jazykov, nahráť tento program priamo do PLC a monitorovať správanie ovládaného stroja. Niektoré softwary umožňujú taktiež vytvárať užívateľské rozhranie (HMI - human machine interface) pre jednoduchšie monitorovanie a riadenie procesu.

3.7.1 Siemens TIA Portal

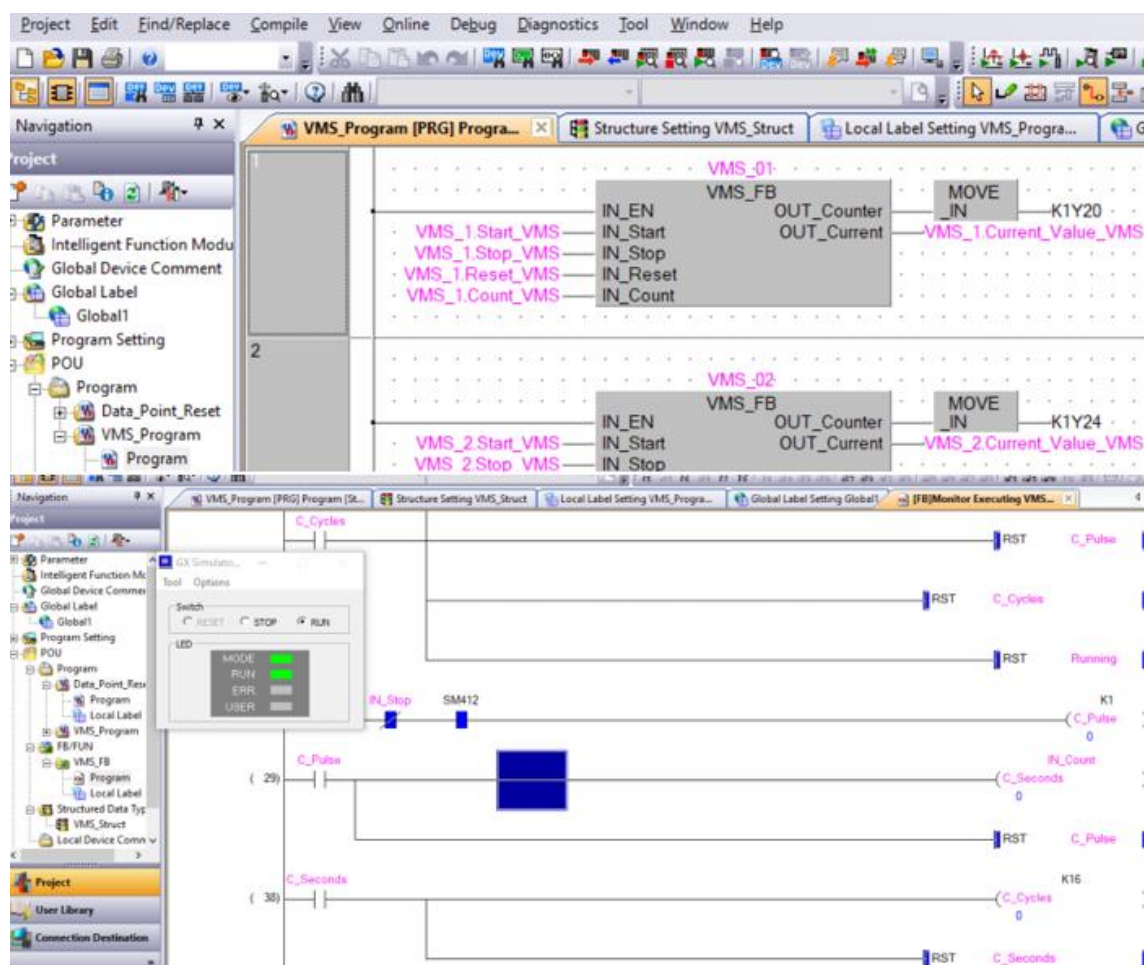
Ďalším nemeckým zástupcom je TIA Portal (Totally Integrated Automation Portal) od firmy Siemens. Tento program, obr. 6), má spoločné prostredie pre zjednotené periférie PLC, pre aplikačnú časť, kde sa tvoria programy, ktoré riadia PLC, a pre vytváranie HMI, čo znamená vytváranie užívateľského rozhrania pre monitorovanie a riadenie PLC. Nájdeme tu aj dispečerské SCADA systémy či pohony zo skupiny Sinamics. Všetky časti sú spolu prepojené aby uľahčili prácu, ktorá je spojením všetkých troch celkov. Taktiež obsahuje veľa preddefinovaných nástrojov a knižníc, aby uľahčili prácu programátorovi. Na druhú stranu nie je možné vytvárať si všetko v podstate od základov ako umožňuje GX Works [7].



Obr. 6) Ukážka z programu Siemens TIA Portal

3.7.2 Mitsubishi GX Works2

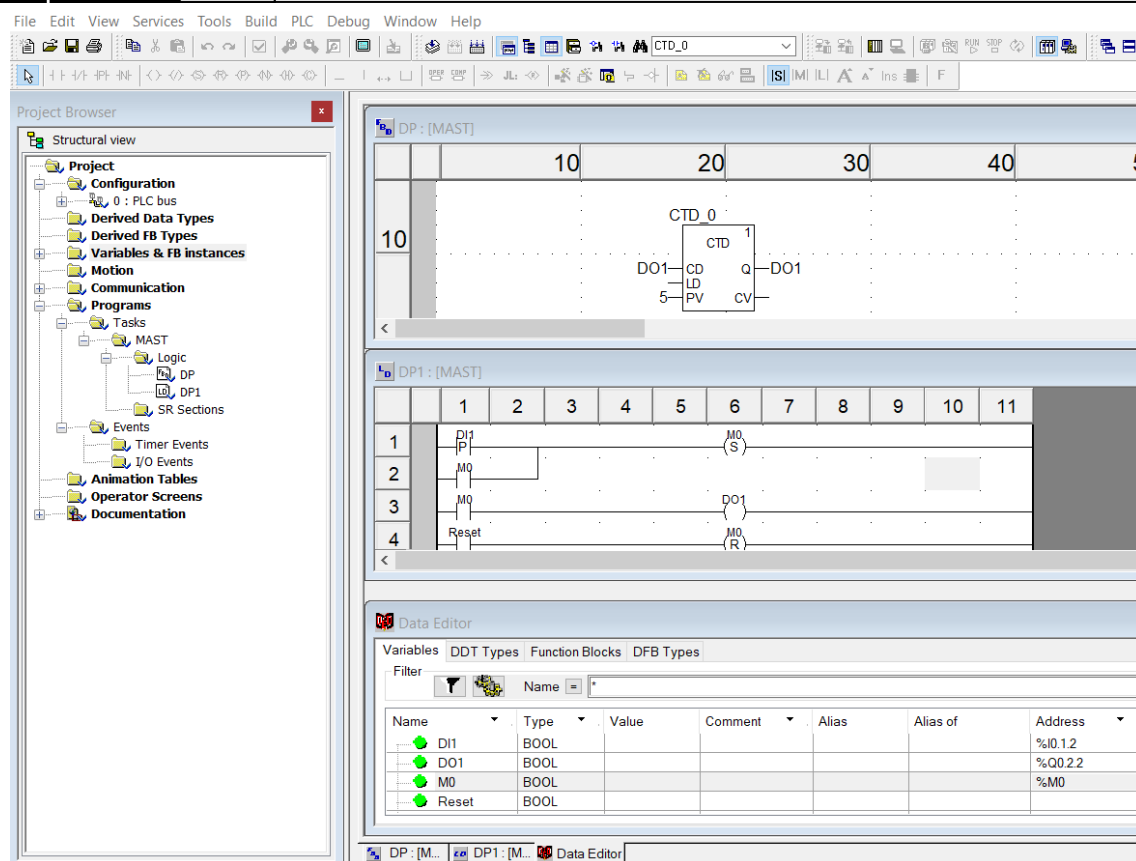
GX Works2 od japonského výrobcu Mitsubishi Electric je prispôsobený modulárnemu systému Q (Q mode) a radám FX a L. Projekt môžeme vytvárať v Simple project pomocou jazykov LD a SFC alebo v Structural project v jazykoch SFC, ST, LD, FBD ako na obr. 7). Tento program je o niečo zložitejší ako vyššie spomínané alternatívy ale na druhú stranu vieme projekt dôvernejšie priblížiť k predstavám používateľa. Program ponúka aj možnosť simulácie vytvoreného programu aj bez reálne pripojeného zariadenia PLC. Programátor tu nájde veľa knižníc funkcií ale aj priestor pre vytvorenie nových [8].



Obr. 7) Ukážka z programu GX Works

3.7.3 Schneider UNITY Pro / EcoStructure Control Expert

Vývojové prostredie ako je na obr. 8) od nemeckej firmy Schneider Electric ponúka inžiniersky software pre PLC ako Modicon M340, Modicon Premium či Quantum. Tieto PLC sú z rady Schneider, a software je využívaný po celú dobu životného cyklu, od návrhu, vývoja až po prevádzkové riadenie. Programovať je možné v LD, FBD, ST, SFC a IL podľa normy IEC61131-3. Po výbere vhodných vstupných a výstupných kariet priamo v Unity Pro, nasleduje výber vhodného napájacieho zdroju, ktorý ukazuje spotrebu vybraných I/O kariet. Premenné je možné jednoducho vytvárať a priamo adresovať na I/O karty. Napríklad digitálny vstup DI2 sa adresuje ako %I0.1.2, t.j. digitálny vstup na nultom (prvom) racku, prvá karta v poradí (čísľujú sa od 0) a druhý channel (čísľujú sa od 0). Po návrhu programu je možné ho odladiť v simulátore, ktorý je tiež zakomponovaný v tomto SW. Navyše je tu možnosť vytvoriť si rôzne online animácie, či zobrazit priebehy analógových premenných. Na obr. 8) sú zobrazené programy vo FBD, LD a premenné.

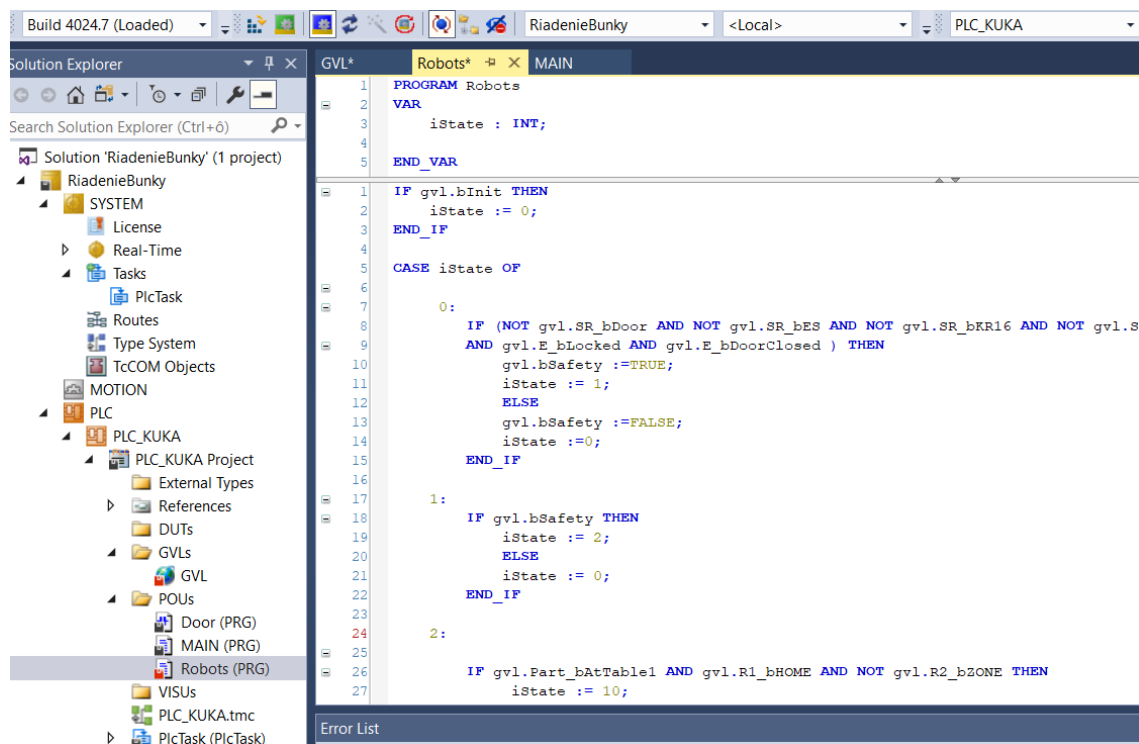


Obr. 8) Ukážka z programu Unity Pro

3.8 Beckhoff TwinCAT3

„The Windows and Control Automation Technology“, skratene TwinCAT je dielom nemeckého výrobcu PLC Beckhoffu. Tento software pracuje na pozadí systému PC windows, pretože pri inštalácii sa zapísal do koreňovej zložky operačného systému, aby si mohol v prípade potreby vyhradiť jadro processoru pre svoje výpočty. V TwinCAT2 bolo prostredie rozdelené na dva celky – PLC control, kde sa navrhuje program a časť system manager kde sa nastavujú komunikácie. V tejto novšej verzii je všetko spolu pod záštitou Visual studio shell. Tu sa tvorí PLC program programovacími jazykmi ST, IL, LD, FBD a SFC, ktoré budú predstavené ďalšej kapitole.

Taktiež sa tu nastavujú zbernice, komunikačné protokoly, vstupné, výstupné a iné rozširovacie moduly. TwinCAT3 komunikuje cez zbernice Interbus, Ethernet, EtherCAT, PROFIBUS DP/MC, CANopen, SERCOS, Lightbus a PC hardware ako USB, sériový či paralelný port [6]. Obr. 9) ukazuje vývojové prostredie programu TwinCAT.



Obr. 9) Ukážka z TwinCAT

3.8.1 Programové prostredie v TwinCAT

Vývojové prostredie programu TwinCAT sa skladá zo 7 základných častí. Menovite sú to „solution explorer“, „editor window“, „message window“, „toolbox“, „properties window“, „toolbars“ a „menu bar“. Tieto okná a jednotlivé toolbars je možné prispôbiť vlastným potrebám. Medzi toolbars by nemali chýbať TwinCAT PLC (Run, Stop, Login, Logout), TwinCAT XAE (Aktivácia konfigurácie, výber cieľu či run mode / free mode - tieto módy budú vysvetlené neskôr). V prípade potreby vizualizácie je tu toolbar TwinCAT PLC Visualization, kde je možné editovať HMI. TwinCAT môže bežať v „Local“, tzn. XAE (engineering) nevytvára spojenie s XAR (run-time), bežia súčasne na PC, ale nekomunikujú spolu. Pre ladenie PLC použijeme PLC run-time, kde bude nastavené „Local“, následne sa pripojí cez príkaz Login a Run PLC.

V prípade, že bude program odladený a pripravený na aplikovanie do PLC v robotickej bunke na škole, bude použité pole „Target“, kde sa pripojíme k cieľovému zariadeniu pomocou ciest („Routes“), cez ktoré sa komunikuje ethernetovým či wi-fi spojením.

Vyššie spomínané režimy run móde, free mode a konfiguračný mód sa nastavujú v hornej časti obrazovky. Run móde je nevyhnutný pre RUN PLC, avšak neznamená to, že PLC je v „Run“. Config móde znamená, že PLC nebeží a EtherCAT neprenáša žiadne informácie. V tomto móde sa vykonávajú zmeny v programe či samotné programovanie. Free run je častou config módu, kde sú už procesné dáta ako vstupy a výstupy prenášané a dajú sa meniť. EtherCAT prenáša dáta nezávisle od PLC cez PLC Idle task, ktorý beží na pozadí. V prípade, že ikonka config / run mode svieti na červeno, znamená to, že je TwinCAT v stop móde alebo XAR detekovalo chybu. Žltá farba indikuje chybu, ktorú prehliadač nie je schopný detekovať napríklad delenie nulou. Posledná šedá farba hovorí, že runtime nie je aktívny.

3.8.2 Vytvorenie PLC projektu

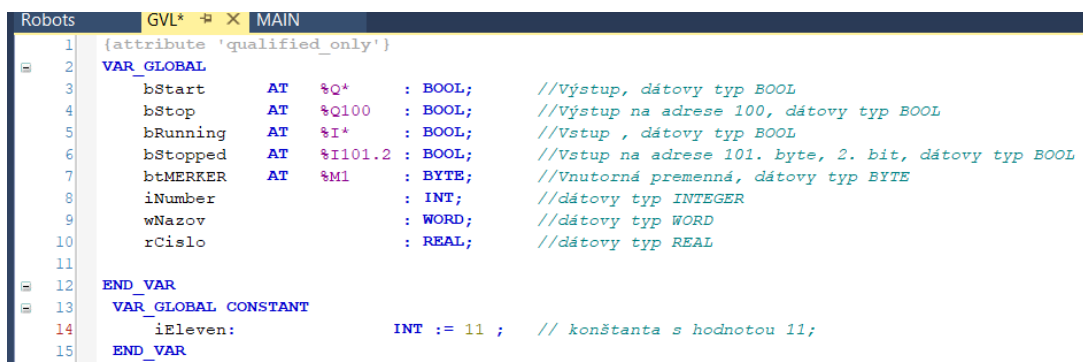
V „solution explorer“ po kliknutí na PLC sa vytvorí projekt PLC. Standard PLC projekt vytvorí kompletnú štruktúru vrátane:

- REFERENCES - knižnice,
- MAIN – PLC volanie
- POU – Programová organizačná jednotka
- GVL - Globálny list premenných
- VISU- vizualizácia.

Pre EtherCAT komunikáciu je nutné mať nainštalovaný vhodný driver. Tu je možnosť zariadenia nájsť automaticky pomocou scanu ktorý je možný len v konfiguračnom móde ale aj ručne pomocou sprievodcu. Po nastaveniach v linkovaniach, PLC Tasku a konfigurácii terminálov je treba konfiguráciu aktivovať ikonkou „Activate configuration“ vedľa ikoniek na spomínané režimy. Celý postup nahrávania programu bude zobrazený v kapitole implementácie softwaru.

3.8.3 Deklarácia premenných

V záložke GVL (Global Variable List) sa deklarujú globálne premenné. Globálne znamená, že premenné budú vidieť vo všetkých častiach PLC projektu. Ak chceme použiť tieto premenné v projekte je nutné použiť „predvoľbu gvl.“ napr. gvl.NazovPremennej. Je taktiež možné tu deklarovať aj konštantu. Na obr. 10) sú zadeklarované najpoužívanéjšie dátové typy. Ak je použitý znak „*“, znamená to že adresa bude určená systémom a nie užívateľom.

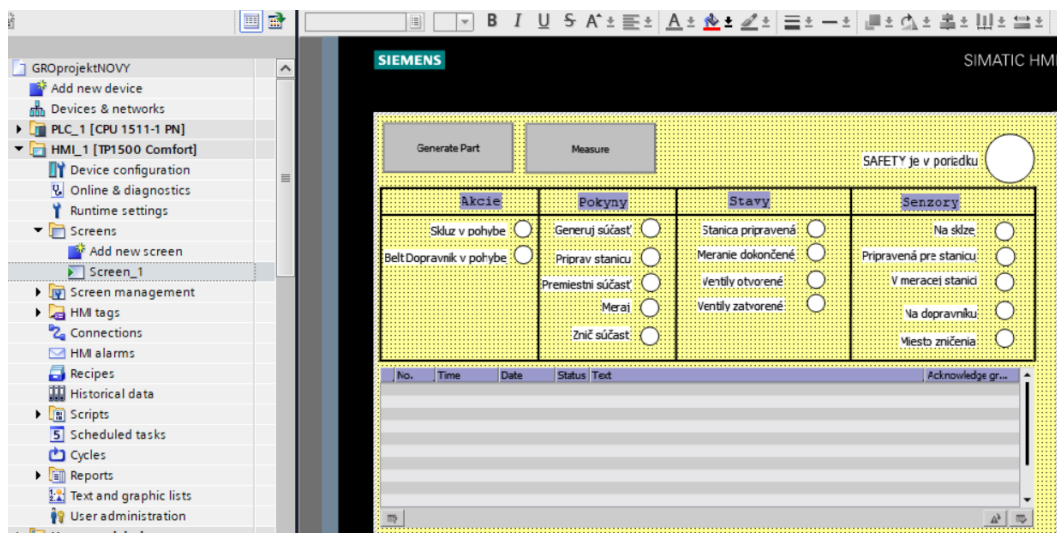


Obr. 10) Premenné v GVL

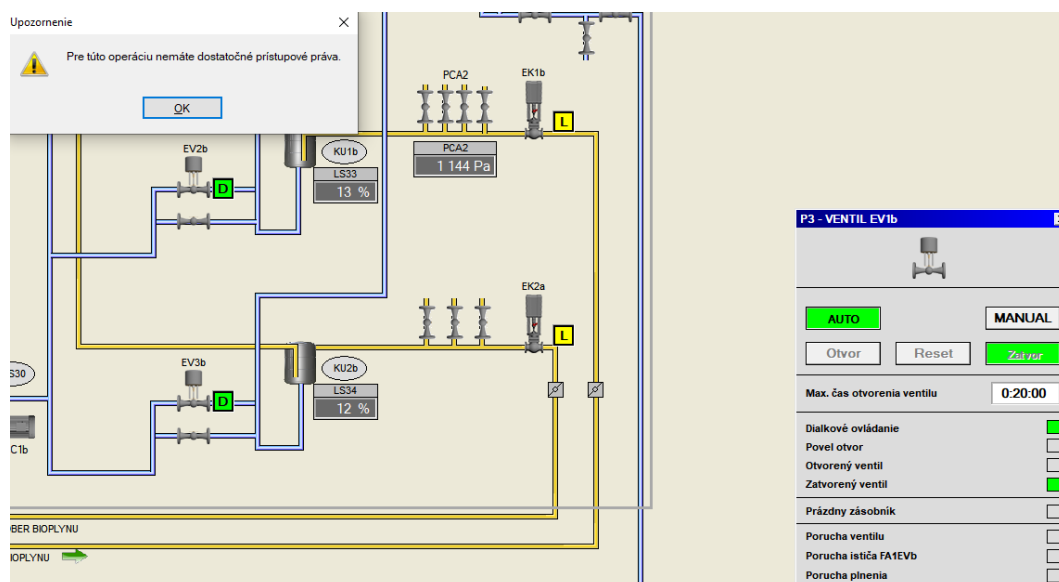
Premenné, ktoré sa zadeklarujú v jednotlivých POU, je možné používať aj v iných POU, aj keď sú iba lokálne, ale je možné ich len čítať, nie do nich zapisovať a je potrebné použiť predvoľbu s názvom POU, kde boli vytvorené napr. MAIN.DO3.

3.9 Software na vytváranie užívateľského rozhrania

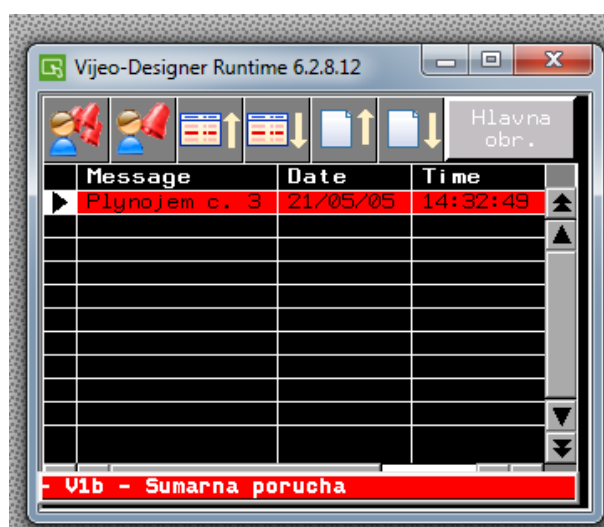
TwinCAT a TIA portal (obr. 11) obsahujú aj časť vizualizácie, kde sa dajú vytvárať HMI ovládacie panely bez potreby ďalšieho softwaru. Mitsubishi GX Works a Schneider Unity Pro potrebujú ďalší software pre vizualizácie ako napríklad Reliance – obr. 12), Vijeo – obr. 13) a mnoho ďalších.



Obr. 11) Vytváranie HMI v TIA portal



Obr. 12) Vizualizácia v programe Reliance



Obr. 13) Vizualizácia v programe Vijeo

4 VYTVORENIE SW PRE RIADENIE ROBOTICKEJ BUNKY

Vhodný software nepochybne patrí medzi najdôležitejšie aspekty výrobného pracoviska či robotickej bunky. Práve tu sa dejú všetky pozorovacie a rozhodovacie funkcie a následne vydávanie povelov pre akčné členy. V tejto kapitole bude na začiatok bližšie popísaný program TwinCAT, ktorý bude tvoriť PLC riadenie. Následne budú popísane stavové automaty riadenia a jednotlivé programy. Posledná časť bude venovaná programu WorkVisual, kde sú vytvorené programy pre riadenie jednotlivých robotov.

4.1 TwinCAT

TwinCAT sa rozdeľuje na dve základné celky:

- **System manager** – konfigurácia komunikácie medzi počítačom a PLC. Tu sa nastavujú vstupy, výstupy, zbernice na PLC a iné karty
- **PLC controller** – programovanie riadiaceho programu. Ponúka aj možnosť simulácie deju, spúšťať či zastavovať program, meniť parametre a premenné a pozorovať správanie sa systému. V tejto časti je možné vytvoriť aj vizualizácie, ktoré predstavujú reálne riadenie v čase simulovaného či riadeného procesu v prívetivej forme pre užívateľa. Využívané sú rôzne prepínače, tlačidlá, signálové kontrolky a mnoho iného.

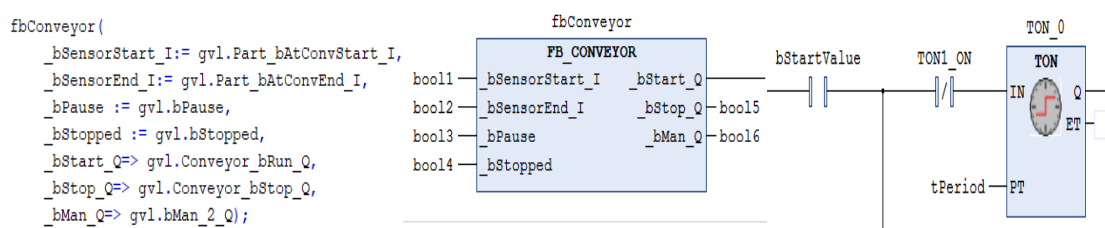
4.1.1 Programovacie jazyky v TwinCAT

The Windows and Control Automation Control, skrátené TwinCAT je run-time program slúžiaci pre PLC a IPC od nemeckej firmy Beckhoff. Programovať je možné v 5 hlavných programovacích jazykoch.

ST – (Structural text) je programovanie po riadkoch v štruktúrovanom texte (obr. 14 a)) písanom programátorom. Patrí medzi vyššie formy programovacích jazykov a jeho základy majú pôvod v Pascale a jazyku C. Na vhodnosti ubera nižšia miera prehľadnosti logických operácií a nutnosť poznať syntax (skladbu jazyka)[2] .

FBD – (Function block diagram) označuje programovanie prostredníctvom funkčných blokov. Tento jazyk funguje na princípe skladania funkčných blokov, ktoré obsahujú už vstavanú logiku, alebo užívateľom vytvorenú logiku do sústavy medzi sebou prepojených blokov. Medzi najznámejšie logické bloky patria AND, OR či TON, TOF ako na obr. 14 b) [2].

LD – (Ladder diagram) má základ v reléovej logike. Pozostáva zo vzájomne pospájaných spínacích kontaktov relé, ktoré vyjadrujú vstupné signály ako na obr. 14 c). Výstupné signály reprezentujú cievky [2].



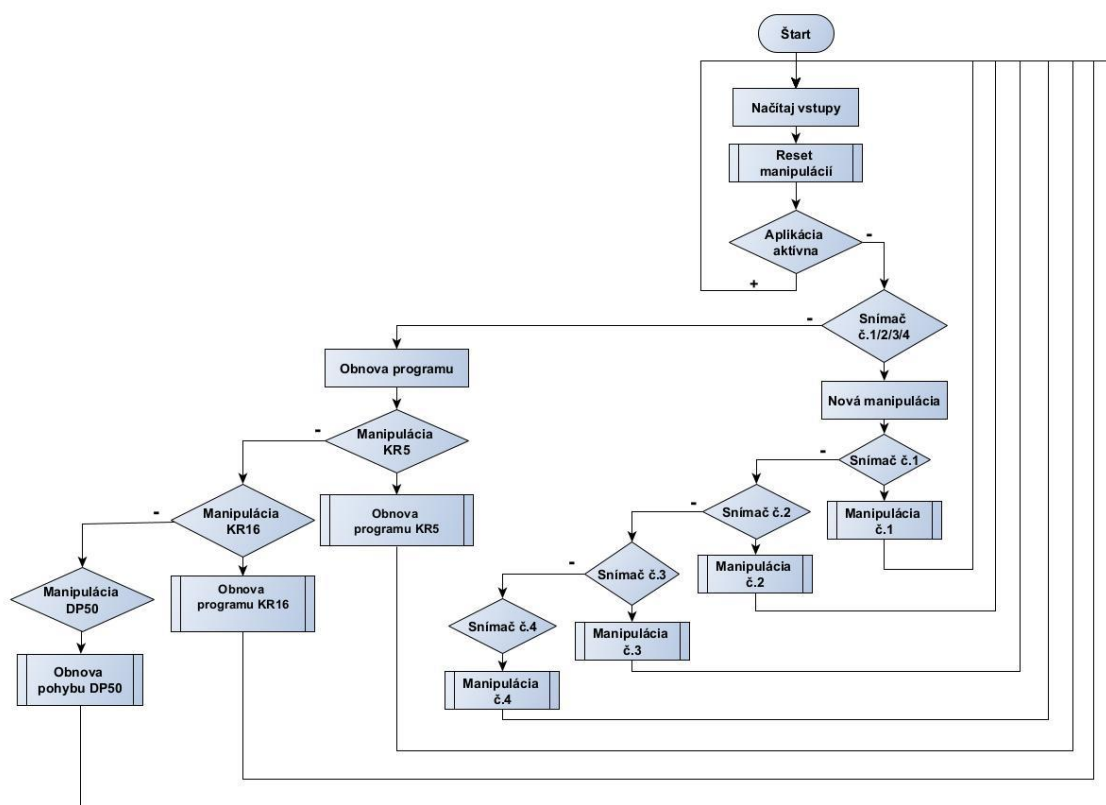
Obr. 14) a) ST, b) FBD, c) LD

II – (Instruction list) má skladbu jazyku podobnú skladaniu jednoduchých inštrukcií, ktoré tvoria celky operácií. Medzi výhody patrí možno najväčšia kontrola nad programom z pohľadu programátora a možnosť vytvorenia programu bez zbytočných volaní funkcií a odskokov [2].

SFC – (Sequential function chart) je graficky orientovaný program podobne ako FBD či CFC. Tento jazyk využíva schému vetvenia programu na základe splnených podmienok. Použitie nájdeme hlavne pri vytváraní sekvenčnej logiky a pri postupnom volaní podprogramov, ktoré mohli byť vytvorené aj v iných programovacích jazykoch [2].

4.1.2 Základný stavový automat PLC

Jednotlivé roboty v tejto práci budú riadené pomocou programu externej automatiky (bude vysvetlený neskôr), kde bude stavový automat, ktorý bude riadený ďalším ale už nadriadeným stavovým automatom z PLC. Dopravník bude povelovaný taktiež z PLC stavového automatu tak ako roboty, ktorého vývojový diagram je znázornený na obr. 15)



Obr. 15) Vývojový diagram riadenia robotov

PLC stavový automat bude riadený logikou pomocou signálov z oboch robotov, dopravníku, ovládacieho panelu a HMI vizualizácie. Task s riadiacim PLC stavovým automatom bude bežať paralelne s taskom, ktorý bude zabezpečovať a čo najvernejšie odrážať všetky safety prvky reálnej praxi – skrátka bezpečnosť.

V jednotlivých funkčných blokoch sa priradujú číselné hodnoty pre alarmy (*nErrorID*) a statusy (*nStatusID*). Každá priradená číselná hodnota má svoju textovú podobu uloženú v *ErrorListe* , poprípadе v *StatusListe*. Tieto textové hodnoty sa následne zobrazujú vo vizualizácii aby pomohli užívateľovi určiť čo sa deje v programe alebo nájsť miesto výskytu alarmovej udalosti. Ukážka z *ErrorListu* je na obr. 16).

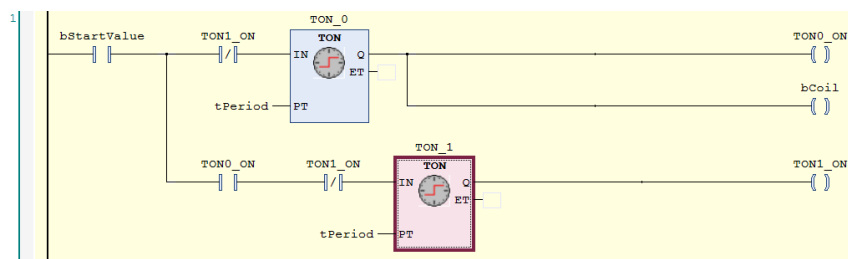
explorer (Ctrl+o)	0	Všetko je OK
References	13	Viac krat sa nepodarilo s... podmienky - bPERI_R...
DUTs	15	Viac krat sa nepodarilo ...ť podmienky - bPRO_A...
E_ManSTATE (ENUM)	17	Viac krat sa nepodarilo spniť podmienky - bAPPL_...
E_RobotSTATE (ENUM)	32	Viac krat sa nepodarilo spniť podmienky - bPERI_...
GVLs	50	PRERUŠENE SAFETY - STOP tlačítko
ErrorList	51	PRERUŠENE SAFETY - E-STOP
GVL	52	PRERUŠENE SAFETY - OPLOTENIE
GVL_HMI	53	PRERUŠENE SAFETY - SVETELNA ZAVORA
StatusList	54	PRERUŠENE SAFETY - KEY SWITCH SA3 R1
POUs	55	PRERUŠENE SAFETY - KEY SWITCH SA3 R2
Blik (FB)	56	PRERUŠENE SAFETY - ZAMOK EUCHNER
DOOR (PRG)	70	PRERUŠENE SAFETY - MENIČ DOPRAVNÍK
FB_CASOVAC (FB)	1	PRERUŠENE SAFETY

Obr. 16) ErrorList

4.1.3 Podporné funkčné bloky

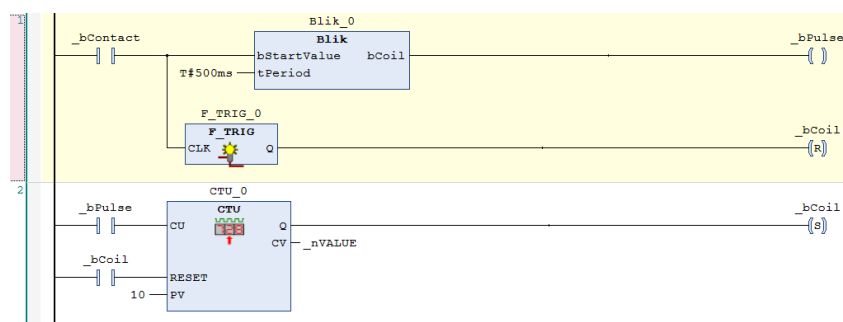
Kvôli jednoduchšiemu a názornejšiemu programu som sa rozhodol pre vytvorenie jednoduchých funkčných blokov, ktoré budú volané pomocou inštancií a používané v programe viacej krát.

Blok *Blik* má za úlohu generovať prerušovaný signál na výstupe s určitou periódou, keď sa mu zopne vstupné relé. Pre názornosť je použitý napríklad na blikanie kontroliek. Ak príde signál na rozblikanie, pomocou 2 časovačov s rovnakou a užívateľom nastavenou periódou budú striedavo zapínať a vypínať výstup. Napríklad ak budú mať časovače periódou 500 ms, na výstupe sa bude každú sekundu meniť hodnota z *High* na *Low* a naopak. Funkčný blok *Blik* je na obr. 17).



Obr. 17) Funkčný blok Blik

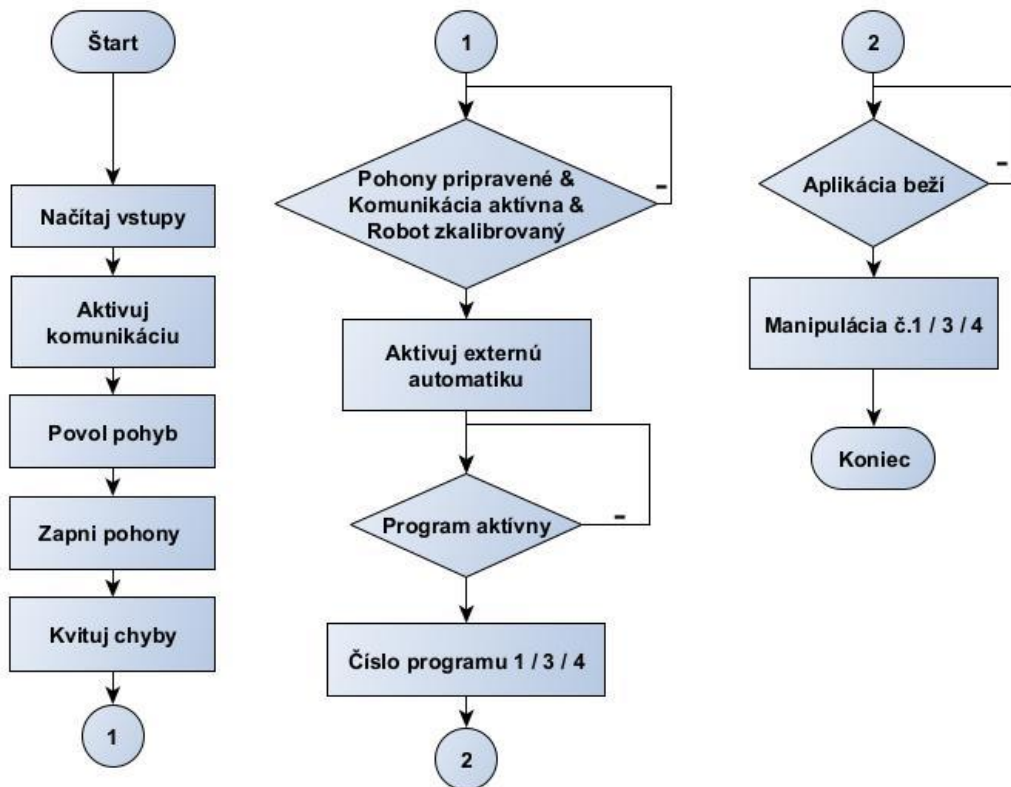
Druhým vytvoreným blokom, obr. 18), je blok časovač, ktorý má podobný princíp ako obyčajný vstavaný TON v TwinCAT, len s rozdielom, že TON si po presušení signálu nepamätá hodnotu, kde sa zastavil. Z toho dôvodu som použil funkčný blok Blik, ktorý mi generuje signál, ktorý reprezentuje sekundové intervaly, ktoré sú počítane čítačom. Po presušení vstupného kontaktu si čítač pamätá hodnotu, kde bol prerušený. Využitie je pri dopravníku, aby čo najvernejšie simuloval snímač na konci dopravníku vzhľadom na dobu, po akú je dopravník spustený.



Obr. 18) Funkčný blok Časovač

4.1.4 FB_ROBOT

Vo funkčnom bloku FB_ROBOT sa aktivuje komunikácia s kontrolérom robota a spúšťa program. V tomto funkčnom bloku je použitý ďalší stavový automat s premennou *_eSTATE*, ktorá je typu *E_RobotSTATE (ENUM)*. ENUM je dátový typ, ktorý slúži k vyjadreniu číselných hodnôt pomocou textu. Vďaka tomu sú stavy viac prehľadné a vypovedajúce o ich obsahu viac ako len prosté čísla, aj keď je možné používať aj čísla príslušného ENUMU namiesto textu. Vývojový diagram pre tento funkčný blok je vidieť na obr. 19).



Obr. 19) Vývojový diagram pre funkčný blok *FB_ROBOT*

V prvom stave *INIT* sa pošlú do kontroléru robota nasledovné inštrukcie:

- Zapni komunikáciu, t.j. *_bI_O_ACT_Q := TRUE;*
- Povoľ pohyb Robotu, t.j. *_bMOVE_ENABLE_Q := TRUE;*
- Zapni pohony, t.j. *_bDRIVES_ON_Q := TRUE;*
- Kvitovanie chybových stavov len na dobu 2 s, pomocou funkcie TP (časový pulz), t.j. *TP_0(IN:= TRUE, PT:= T#2S, Q=> _bCONF_MESS_Q, ET=>);*

Potom sa nastaví automat na hodnotu *DRIVES*. V tomto stave sa kontrolujú podmienky:

- a) Sú pohony pripravené? *_bPERI_RDY_I = TRUE?*
- b) Je aktívne rozhranie ? *_bIO_ACT_CONF_I = TRUE?*
- c) Sú osi robota zkalibrované? *_bROB_CAL_I = TRUE?*

V prípade ak aspoň 1 z podmienok nie je splnená, *_eSTATE* dostane znovu hodnotu *DRIVES*, t.j. znova sa skontrolujú podmienky v tomto stave. Ak by sa niekoľko krát po sebe nepodarilo tieto podmienky splniť, nastaví sa stavový automat na hodnotu *INIT*, aby sa všetky príkazy znova zopakovali a taktiež sa nastaví príslušná hodnota *nErrorID* a tá odošle alarm do

vizualizácie. Rovnaký spôsob odozvy bude v každom zo stavov *FB_ROBOT*, preto už nebude ďalej rozoberaný. Ak sú splnené všetky podmienky *_eSTATE* sa nastaví na hodnotu *START_EXT*. Na obr. 20) je zobrazený funkčný blok *FB_ROBOT*.

```

8  CASE _eSTATE OF
9      E_RobotSTATE.INIT:
10         _nStatusID := 11;
11         _bIO_ACT_Q := TRUE;
12         _bMOVE_ENABLE_Q := TRUE;
13         _bDRIVES_ON_Q := TRUE;
14         _bCONF_MESS_Q := TRUE;
15         TP1(IN:= TRUE, PT:= T#2S, Q=> _bCONF_MESS_Q, ET=> );
16         iii := 0;
17         _eSTATE := E_RobotSTATE.DRIVES;
18     E_RobotSTATE.DRIVES:
19         IF _bPERI_RDY_I AND _bIO_ACT_CONF_I AND _bROB_CAL_I THEN
20             i:=0;
21             _nStatusID := 12;      _nErrorID := 0;
22             _eSTATE := E_RobotSTATE.START_EXT;
23         ELSE
24             i:=i+1;
25             IF i = 500 THEN
26                 _nErrorID := 13;
27                 i:=0;
28                 _eSTATE := E_RobotSTATE.INIT;
29             END_IF
30             _nStatusID := 13;
31             _eSTATE := E_RobotSTATE.DRIVES;
32         END_IF
33     E_RobotSTATE.START_EXT: [15 lines]
34     E_RobotSTATE.NUMBER: [19 lines]

```

Obr. 20) Funkčný blok v PLC programe

Stav *START_EXT* začína aktivovaním externej automatiky na riadenie robota; *_bEXT_START_Q := TRUE* a pokračuje čítaním hodnôt z robota a porovnáva ich s požadovanými hodnotami pre ďalší vývoj operácie. Ak je aktívny program *CELL.SRC* (bude vysvetlený neskôr) , t.j. *_bPRO_ACT_I = TRUE* nastaví sa stav *NUMBER*.

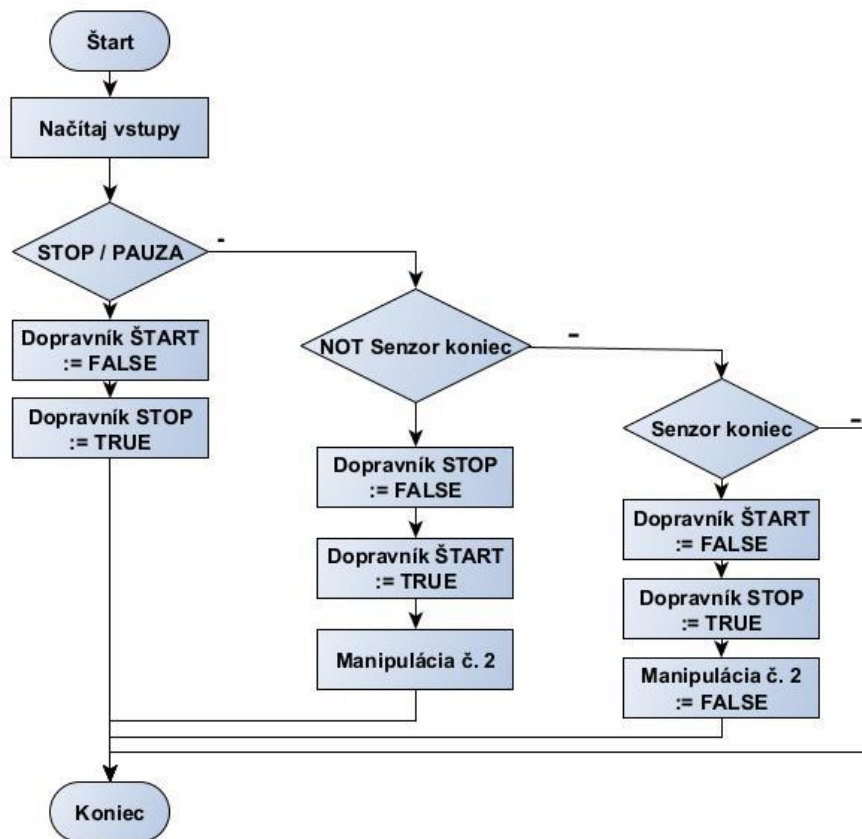
V poslednom stave *NUMBER* sa vykoná nasledovný príkaz:

- Priradiť číslo programu ku robotu, t.j. *_nPGNO_Q := 1 / 3 / 4*;

Pokiaľ beží program na manipuláciu (robot vykonáva manipulačnú operáciu), *_bAPPL_RUN_I = TRUE*, nastaví sa, že manipulácia je aktívna; *_bMan_Q := TRUE* a priradia sa odpovedajúce čísla statusu či alarmu. Následne sa *_eSTATE* nastaví na hodnotu *INIT*, čo je východzí stav automatu.

4.1.5 FB_CONVEYOR

V tomto funkčnom bloku sú vystavené tri podmienky. Ak nie je súčasť na konci, dopravník bude uvedený do pohybu nastavením premennej *_bStart_Q := TRUE* a *_bStop_Q := FALSE*. Táto premenná vyšle signál do meniču, ktorý prostredníctvom motoru uvedie pás na dopravníku do pohybu. Následne sa *_bMan_Q* nastaví do hodnoty *TRUE* a celý cyklus sa opakuje. V momente ako je potvrdená prítomnosť súčiastky na konci dopravníku snímačom, dopravník je zastavený zhodením signálu pre pohyb dopravníku; *_bStart := FALSE* a *_bStop_Q := TRUE*, manipulácia sa vyresetuje; *_bMan_Q := FALSE*. Vývojový diagram je funkčného bloku na obr. 21) a jeho PLC program na obr. 22).



Obr. 21) Vývojový diagram pre pohyb dopravníka

```

1  IF NOT _bSensorEnd_I AND NOT _bPause AND NOT _bStopped THEN
2      _bStop_Q := FALSE;
3      _bStart_Q := TRUE;
4      _bMan_Q := TRUE;
5      _nStatusID := 20;
6  ELSIF _bSensorEnd_I THEN
7      _bStart_Q := FALSE;
8      _bStop_Q := TRUE;
9      _bMan_Q := FALSE;
10     _nStatusID := 21;
11  ELSIF _bPause OR _bStopped THEN
12      _bStart_Q := FALSE;
13      _bStop_Q := TRUE;
14      _nStatusID := 22;
15  END_IF

```

Obr. 22) Funkčný blok *FB_CONVEYOR*

4.1.6 FB_RECOVERY

Tento funkčný blok slúži na opätovné spustenie programu a robota do pohybu. Jeho vývojový diagram je na obr. 23). V ňom sa na začiatku pošlú signály pre zapnutie pohonov a signál kvitovania chyb;

_bDRIVES_ON_Q := TRUE;

_bCONF_MESS_Q := TRUE;

Následne sa po prejdení 500 ms kontroluje podmienka, či sú pohony pripravené (*_bPERI_RDY_I = TRUE*), je stlačené štart tlačítko (*_bStart = TRUE*), a je dodržaná bezpečnosť obsluhy (*_bUSER_SAF_I = TRUE*). Ak sú všetky vyššie spomínané podmienky

splnené, premenná *_bEXT_START* sa nastaví do hodnoty *TRUE*, čím spustí program externej automatiky a program pokračuje tam, kde bol prerušený.



Obr. 23) Vývojový diagram funkčného bloku *FB_RECOVERY*

4.1.7 FB_RESET_MAN

Funkčný blok slúži na resetovanie premenných, ktoré vyjadrujú, aká manipulácia sa momentálne vykonáva alebo bola prerušená. Program pozostáva z troch jednoduchých podmienkových konštrukcií a to, či sa príslušný robot nachádza v Home-pozícii a manipulovaná súčasť na mieste určenia. Náležiaci program je na obr. 24).

```

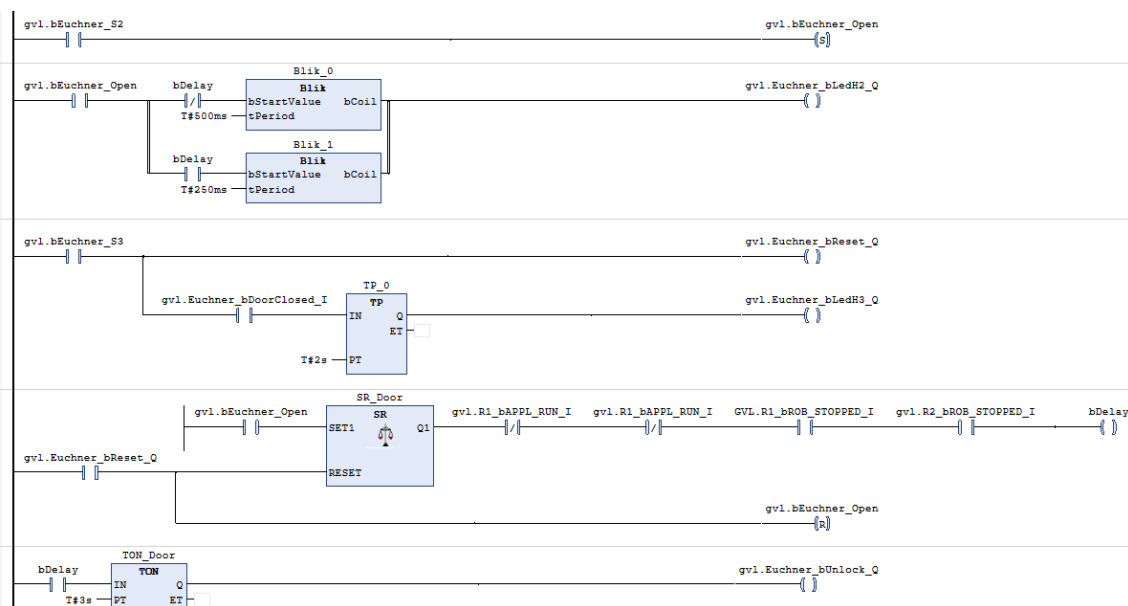
1  IF _bR1_IN_HOME_I AND _SensorStart_I THEN
2      _bMan1_Q := FALSE;
3      _nR1_PGNO_Q:= 0;
4  END_IF
5
6  IF _bR1_IN_HOME_I AND _SensorT1_I THEN
7      _bMan4_Q := FALSE;
8      _nR1_PGNO_Q:= 0;
9  END_IF
10
11 IF _bR2_IN_HOME_I AND _SensorT2_I THEN
12     _bMan3_Q := FALSE;
13     _nR2_PGNO_Q:= 0;
14 END_IF
  
```

Obr. 24) Funkčný blok *FB_RESET_MAN*

4.1.8 Program Door

Nasledujúci program slúži na ovládanie bezpečnostného zámku od firmy Euchner. Pri stlačení horného tlačítka na zámku sa odošle požiadavka na otvorenie zámku. Počas trvania tejto požiadavky bliká podsvietenie tlačítka s periódou 1 s. Dvere je možné otvoriť ak nie je aktívna žiadna robotizácia aplikácia, oba roboty stoja a tieto stavy trvajú aspoň 3 s. Počas tejto doby

začne podsvietenie tlačítka blikať rýchlejšie a to s periódou 500 ms. Potom sa dvere odistia a je možné ich otvoriť. Ak chceme dvere zatvoriť je potrebné stlačiť spodné tlačítko, ktoré resetuje požiadavku na otvorenie. Posvietenie tohto tlačítka sa rozsvieti na 2 s (na bunke bude blikať 2 s), čo symbolizuje zatvorenie a zaistenie dverí. Program v LD jazyku je na obr. 25).

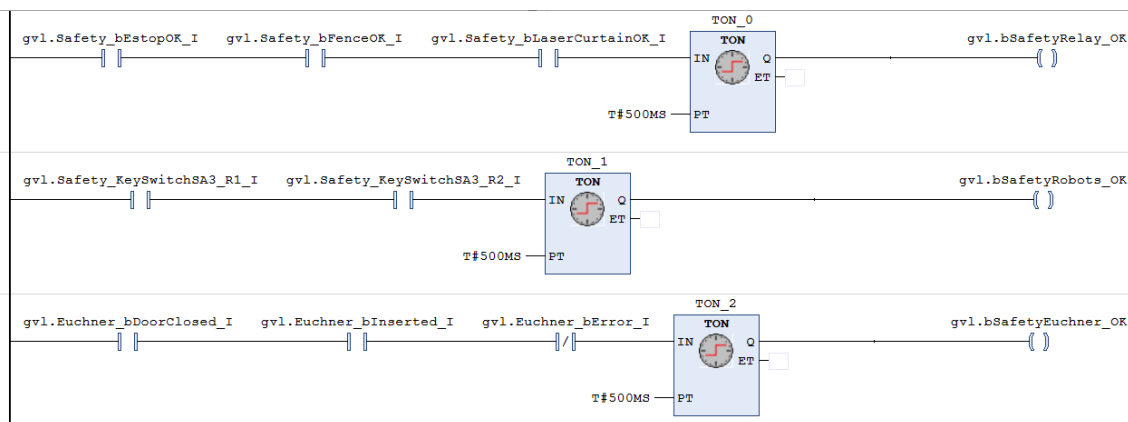


Obr. 25) Program Door

4.1.9 Program Safety

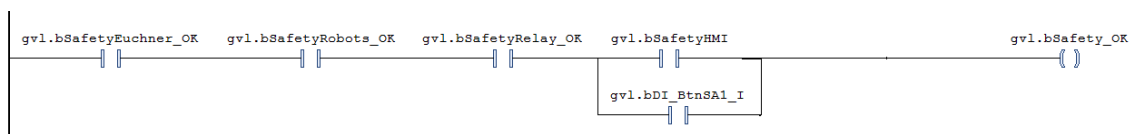
Program safety bol vytvorený v separátnom „tasku“ s názvom *Task_SAFETY*. V reálnej praxi býva bezpečnosť riešená napríklad ďalším PLC a celou radou prídavných zariadení. Bezpečnosť v tejto práci bude len názorná pre priblíženie problematiky, ktorá je mimo rozsah zadania tejto diplomovej práce.

Program je tvorený sériovým pospájaním signálov Emergency stop, oplatenie a svetelná závara. Ak sú všetky 3 v hodnote *High* po dobu 500 ms je nastavená sumárna premenná, ktorá združuje safety relé. Rovnaký postup je zvolený aj pri bezpečnostných prepínačov na paneli a pre signály zo zámku Euchner a všetky sú znázornené na obr. 26).



Obr. 26) Dielče bezpečnosti

Celková bezpečnosť je dosiahnutá v prípade *High* hodnoty sumárnych bezpečností zapojených v sérii a zároveň je stlačené safety tlačítko na paneli alebo vo vizualizácii ako je na obr. 27).



Obr. 27) Celková bezpečnosť

V prípade, že jeden z prvkov bezpečnosti bude zhodený, zastaví sa program a nastaví sa premenná *gvl.bStopped*, ktorá symbolizuje poistku proti opätovnému spusteniu v prípade pominutia nebezpečného stavu. Ak je táto premenná aktívna, je nutné ju resetovať pomocou tlačítka „Štart“ držaného aspoň 3 s vo vizualizácii. Keď je toto tlačítko stlačené, bliká zelená farba aby mal užívateľ odozvu od systému.

4.1.10 Program MAIN

V tejto podkapitole bude podrobne vysvetlená logika PLC stavového automatu.

Samotný projekt obsahuje už preddefinovanú funkciu „MAIN“, ktorú volá PLC pomocou PLC tasku, ktorý má čas cyklu nastavený na 10 ms. *Task_SAFETY* má kvôli bezpečnosti nastavený čas cyklu na 1 ms aby bol schopný prerušiť PLC program včas.

Kvôli prehľadnosti programu a ľahšej manipulácii s ním je zvolená ako hlavnú kostra celého riadenia konštrukcia *CASE OF - END_CASE*. V tomto prevedení je kontrolovaná premenná *eStateMAN*, ktorá je typu *E_ManSTATE* (ENUM) a jej deklarácia je zobrazená na obr. 28).

```

3  TYPE E_ManSTATE :
4  (
5      INIT := 1,
6      CONDITION_EVALUATION := 10,
7      RECOVERY := 20,
8      NEW_MAN := 30,
9      MAN1 := 110,
10     MAN2 := 120,
11     MAN3 := 130,
12     MAN4 := 140
13 ) ;
14 END TYPE
15

```

Obr. 28) ENUM E_ManSTATE

V stavovom automate sa na základe splnených podmienok volia stavy, v ktorých sa vykonávajú inštrukcie a volajú funkčné bloky. Hlavný stavový automat je rozdelený do 8 základných stavov ako je na obr. 28) a obr. 29) .

```

1 DOOR();
2 MANUAL();
3 HMI();
4
5 IF gvl.bSafety_OK AND NOT gvl.bStopped AND NOT gvl.bPause AND NOT gvl.bManual THEN
6
7     CASE eStateMAN OF
8         E_ManSTATE.INIT : [22 lines]
9         E_ManSTATE.CONDITION_EVALUATION: [6 lines]
10        E_ManSTATE.NEW_MAN: [11 lines]
11        E_ManSTATE.MAN1: [27 lines]
12        E_ManSTATE.MAN2: [33 lines]
13        E_ManSTATE.MAN3: [28 lines]
14        E_ManSTATE.MAN4: [28 lines]
15        E_ManSTATE.RECOVERY: [39 lines]
16
17    END_CASE
18 END_IF

```

Obr. 29) Program MAIN

V prvom stave, ktorý nesie názov *INIT*, obr. 30), je volaný funkčný blok *FB_RESET_MAN*, kde sa resetujú jednotlivé manipulácie pokiaľ boli úspešne ukončené splnením koncovej pozície manipulovaného predmetu a návratu do Home-pozície príslušného robota či dopravníku. Nasleduje podmienka, kde sa zisťuje, či je aktívna jedna z manipulácií, nie je spustený manuálny režim (vysvetlenie neskôr) alebo je program zastavený či pozastavený. Ak sú tieto podmienky splnené, znamená to, že sa vykonáva jedna z manipulácií a stavový automat čaká na jej skončenie, program je zastavený kvôli prerušeniu bezpečnosti alebo je spustený manuálny režim. V prípade, že tieto podmienky nie sú splnené, stavový automat sa prepne na stav *CONDITION_EVALUATION*, obr. 31), kde sa kontroluje, či sa nachádza manipulovaná súčasť na jednej z preddefinovaných pozícií alebo zastavila kvôli prerušenej manipulácii. Podľa týchto podmienok sa prepne buď do stavu spustenia novej manipulácie *NEW_MAN*, alebo do stavu obnovy programu *RECOVERY*.

```

7 E_ManSTATE.INIT :
8   gvl.nStatusID := 1;
9   fbResetMan(
10     _bR1_IN_HOME_I:= gvl.R1_bIN_HOME_I,
11     _bR2_IN_HOME_I:= gvl.R2_bIN_HOME_I,
12     _SensorT1_I:= gvl.Part_bAtTable1_I,
13     _SensorStart_I:= gvl.Part_bAtConvStart_I,
14     _SensorT2_I:= gvl.Part_bAtTable2_I,
15     _bMan1_Q=> gvl.bMan_1_Q,
16     _bMan3_Q=> gvl.bMan_3_Q,
17     _bMan4_Q=> gvl.bMan_4_Q,
18     _nR1_PGNO_Q=> gvl.R1_nPGNO_Q,
19     _nR2_PGNO_Q=> gvl.R2_nPGNO_Q,
20     _nErrorID=> gvl.nErrorID,
21     _nStatusID=>gvl.nStatusID);
22
23   IF gvl.R1_bAPPL_RUN_I OR gvl.R2_bAPPL_RUN_I OR gvl.bStopped OR gvl.bPause THEN
24     eStateMAN := E_ManSTATE.INIT;
25   ELSE
26     eStateMAN := E_ManSTATE.CONDITION_EVALUATION;
27   END_IF

```

Obr. 30) Stav INIT

```

30 E_ManSTATE.CONDITION_EVALUATION:
31   IF (gvl.Part_bAtTable1_I OR gvl.Part_bAtConvStart_I OR gvl.Part_bAtConvEnd_I OR gvl.Part_bAtTable2_I) THEN
32     eStateMAN := E_ManSTATE.NEW_MAN;
33   ELSEIF NOT gvl.R1_bAPPL_RUN_I AND NOT gvl.R2_bAPPL_RUN_I AND NOT gvl.Conveyor_bRun_Q THEN
34     eStateMAN := E_ManSTATE.RECOVERY;
35   END IF

```

Obr. 31) Stav CONDITION_EVALUATION

Pokračovať bude jeden z najhlavnejších rozdelením stavov. V *NEW_MAN* sa nastavuje, ktorý robot či dopravník môže konať manipuláciu v danom čase. Tento stav odkazuje na 4

možné pokračujúce stavy a na jeden, ktorý sa vráti automat späť na začiatok v prípade žiadnej nevyhovujúcej podmienky. Aby bola splnená podmienka musia byť splnené jej čiastkové podmienky a, b. Jednotlivé podmienky v *NEW_MAN* :

1. Podmienka
 - a. Manipulovaná súčasť sa nachádza na stole č.1;
gvl.PartAtTable1 = TRUE
 - b. Robot 2 KR16-2 sa nenachádza v kolíznej zóne;
gvl.R2_Zone_Occupied = FALSE
2. Podmienka
 - a. Manipulovaná súčasť sa nachádza na začiatku dopravníka
gvl.PartAtConvStart = TRUE
3. Podmienka
 - a. Manipulovaná súčasť sa nachádza na konci dopravníka
gvl.PartAtConvEnd = TRUE
 - b. Robot 1 KR5 sa nenachádza v kolíznej zóne
gvl.R1_Zone_Occupied = FALSE
4. Podmienka
 - a. Manipulovaná súčasť sa nachádza na stole č.2
gvl.PartAtTable1 = TRUE
 - b. Robot 2 KR16-2 sa nenachádza v kolíznej zóne
gvl.R2_Zone_Occupied = FALSE

Ak je splnená 1. podmienka, *eStateMAN* dostane hodnotu *MAN1*;

Ak je splnená 2. podmienka, *eStateMAN* dostane hodnotu *MAN2*;

Ak je splnená 3. podmienka, *eStateMAN* dostane hodnotu *MAN3*;

Ak je splnená 4. podmienka, *eStateMAN* dostane hodnotu *MAN4*;

Ak nevyhovuje ani jedna podmienka, stavový automat je nastavený na hodnotu *INIT* a začína od začiatku vyhodnocovať podmienky a nastavovať príslušné stavy. Obr. 32) zobrazuje *stav NEW_MAN* v PLC programe. V každej z nových manipulácií *MAN1-4* sa objavuje aj dvojica podmienok, ktoré pomocou príslušných premenných simulujú snímače prítomnosti dielu na určených pozíciách. Prvá IF konštrukcia resetuje predchádzajúcu pozíciu súčiastky a druhá nastavuje novú.

```

E_ManSTATE.NEW_MAN:
    IF gvl.Part_bAtTable1_I AND NOT gvl.R2_bZone_Occupied THEN
        eStateMAN := E_ManSTATE.MAN1;
    ELSIF gvl.Part_bAtConvStart_I THEN
        eStateMAN := E_ManSTATE.MAN2;
    ELSIF gvl.Part_bAtConvEnd_I AND NOT gvl.R1_bZone_Occupied THEN
        eStateMAN := E_ManSTATE.MAN3;
    ELSIF gvl.Part_bAtTable2_I AND NOT gvl.R2_bZone_Occupied THEN
        eStateMAN := E_ManSTATE.MAN4;
    ELSE
        eStateMAN := E_ManSTATE.INIT;
    END IF

```

Obr. 32) Stav *NEW_MAN*

V prípade, že *E_StateMAN* je rovný *MAN1*, začnú sa vykonávať operácie z manipulácie 1 – Robot 1 – KUKA KR5 ARC príde na pozíciu ku stolu č.1, vezme manipulovaný predmet, presunie sa nad začiatok dopravníku, položí predmet a vracia sa do Home pozície. Vývojový diagram bol zobrazený vyššie na obr. 19) . Program pre túto operáciu manipulácia č.1 je na obr. 33).

```

49 E_ManSTATE.MAN1:
50   fbR1_Man1( //Manipulácia 5.1, Robot KR5 presúva manipulovanú súčasť zo stolu 5.1 na začiatok dopravníku
51     _bSensor_I:= gvl.Part_bAtTable1_I,
52     _bPERI_RDY_I:= gvl.R1_bPERI_RDY_I,
53     _bIO_ACT_CONF_I:= gvl.R1_bIO_ACT_CONF_I,
54     _bROB_CAL_I:= gvl.R1_bROB_CAL_I,
55     _bPRO_ACT_I:= gvl.R1_bPRO_ACT_I,
56     _bPGNO_REQ_I:= gvl.R1_bPGNO_REQ_I,
57     _bAPPL_RUN_I:= gvl.R1_bAPPL_RUN_I,
58     _bNumber:= 1,
59     _bIO_ACT_Q=> gvl.R1_bIO_ACT_Q,
60     _bMOVE_ENABLE_Q=> gvl.R1_bMOVE_ENABLE_Q,
61     _bDRIVES_ON_Q=> gvl.R1_bDRIVES_ON_Q,
62     _bCONF_MESS_Q=> gvl.R1_bCONF_MESS_Q,
63     _bEXT_START_Q=> gvl.R1_bEXT_START_Q,
64     _nPGNO_Q=> gvl.R1_nPGNO_Q,
65     _bPGNO_VALID_Q=> gvl.R1_bPGNO_VALID_Q,
66     _bMAN_Q:= gvl.bMan_1_Q,
67     _nErrorID=> gvl.nErrorID,
68     _nStatusID=>gvl.nStatusID);
69   //Reset premennej Súčasť na stole 5.1 a nastavenie súčasti na začiatku dopravníku
70   IF gvl.bR5Taktel THEN gvl.Part_bAtTable1_I:= FALSE; END_IF
71   IF gvl.bConvStart THEN gvl.Part_bAtConvStart_I:= TRUE; nValue := 0; END_IF
72
73   IF gvl.Part_bAtConvStart_I AND gvl.R1_bIN_HOME_I THEN
74     eStateMAN := E_ManSTATE.INIT;
75   ELSE
76     eStateMAN := E_ManSTATE.MAN1;
77   END_IF

```

Obr. 33) Stav manipulácie 1 (*MAN1*)

Správnou funkciu dopravníku zabezpečuje stav *MAN2*. Tohoto stavu je dosiahnuté v prípade, že manipulovaná súčiastka sa nachádza na začiatku dopravníku. Na začiatku je hlavná podmienka, kde sa zisťuje, či je bezpečnostný prvok dopravníku v poriadku ako ukazuje obr. 34). Tento prvok nebol braný medzi celkovú bezpečnosť, preto je potrebné ho zohľadniť v tomto stave. Ak je bezpečnosť dopravníku splnená, nasleduje volanie funkčného bloku *FB_CONVEYOR*.

```

78 E_ManSTATE.MAN2:
79   IF gvl.Safety_bFMIO_I THEN
80     gvl.nErrorID := 0;
81   //Casovacia dopravníku - simulacia snimacov
82   fbCasovacConveyor(_bContact:= gvl.Conveyor_bRun_Q, _bReset := gvl.Part_bAtConvEnd_I, _bCoil=> gvl.bConvEnd, _nVALU
83     gvl.nPoziciaConv := nValue*20;
84
85   IF gvl.Conveyor_bRun_Q THEN gvl.Part_bAtConvStart_I:= FALSE; END_IF
86   IF gvl.bConvEnd THEN gvl.Part_bAtConvEnd_I:= TRUE; END_IF
87
88   //Manipulácia 5.2, Dopravník je v chode, súčasť sa je premiestňovaná zo začiatku na koniec dopravníku
89   fbConveyor(
90     _bSensorStart_I:= gvl.Part_bAtConvStart_I,
91     _bSensorEnd_I:= gvl.Part_bAtConvEnd_I,
92     _bPause := gvl.bPause,
93     _bStopped := gvl.bStopped,
94     _bMan_I := gvl.bMan_2_Q,
95     _bStart_Q=> gvl.Conveyor_bRun_Q,
96     _bStop_Q=> gvl.Conveyor_bStop_Q,
97     _bMan_Q:= gvl.bMan_2_Q,
98     _nErrorID=> gvl.nErrorID,
99     _nStatusID=>gvl.nStatusID);
100
101   //Vyskocenie alebo zotrvanie v stave
102   IF gvl.Part_bAtConvEnd_I AND NOT gvl.bConvEnd THEN
103     gvl.bMan_2_Q := FALSE;
104     eStateMAN := E_ManSTATE.INIT;
105   ELSE
106     eStateMAN := E_ManSTATE.MAN2;
107   END_IF

```

Obr. 34) Stav *MAN2* pre ovládanie dopravníka z PLC

Do stavu *RECOVERY* sa dostane v prípade, že manipulovaná súčasť sa nenachádza nikde v dosahu senzorov, nebežia žiadne robotické aplikácie a dopravník stojí. Tieto všetky podmienky, obr. 35), znamenajú, že sa jedna z manipulácií nevykonala až do svojho konca

z dôvodu nutného zastavenia spôsobeného, prerušením bezpečnosti, alebo užívateľským okamžitým zastavením.

Tento stav pozostáva z troch opytovacích konštrukcií IF-THEN a tu sú ich jednotlivé podmienky a príkazy:

1. Bola naposledy aktívna manipulácia 1 alebo manipulácia 4?;
gvl.bMan1_Q = TRUE or gvl.bMan4_Q = TRUE?
2. Bola naposledy aktívna manipulácia 3?;
gvl.bMan3_Q = TRUE
3. Bola naposledy aktívna manipulácia?;
gvl.bMan2_Q = TRUE

V prípade splnenia podmienok 1 alebo 2 sa volá funkčný blok *FB_RECOVERY*. V prípade, že naposledy bola aktívna manipulácia 2, odošle sa signál do dopravníku pre uvedenie do pohybu a nastaví sa stav *MAN2*;

```

174      E_ManSTATE.RECOVERY:
175      IF gvl.bMan_1_Q OR gvl.bMan_4_Q THEN //Uvedenie Robota KR5 do opätovnej prevádzky ;
176      fbR1_Recovery(
177          _bPERI_RDY_I:= gvl.R1_bPERI_RDY_I,
178          _bStart:= gvl.bStart,
179          _bUSER_SAF_I:= gvl.R1_bUSER_SAF_I,
180          _bMan:= gvl.bMan_1_Q OR gvl.bMan_4_Q,
181          _bDRIVES_ON_Q=> gvl.R1_bDRIVES_ON_Q,
182          _bCONF_MESS_Q=> gvl.R1_bCONF_MESS_Q,
183          _bEXT_START_Q=> gvl.R1_bEXT_START_Q,
184          _nErrorID=> gvl.nErrorID,
185          _nStatusID=> gvl.nStatusID);
186
187      IF gvl.bMan_1_Q THEN
188          eStateMAN := E_ManSTATE.MAN1;
189      ELSIF gvl.bMan_4_Q THEN
190          eStateMAN := E_ManSTATE.MAN4;
191      END_IF
192      ELSIF gvl.bMan_3_Q THEN //Uvedenie Robota KR16-2 do opätovnej prevádzky
193      fbR2_Recovery( [12 lines]
194
195      ELSIF gvl.bMan_2_Q THEN //Uvedenie dopravníku opätovnej prevádzky
196          gvl.Conveyor_bStop_Q := FALSE;
197          gvl.Conveyor_bRun_Q := TRUE;
198          eStateMAN := E_ManSTATE.MAN2;
199      ELSE
200          eStateMAN := E_ManSTATE.INIT;
201

```

Obr. 35) Obnova programu

4.2 WorkVisual

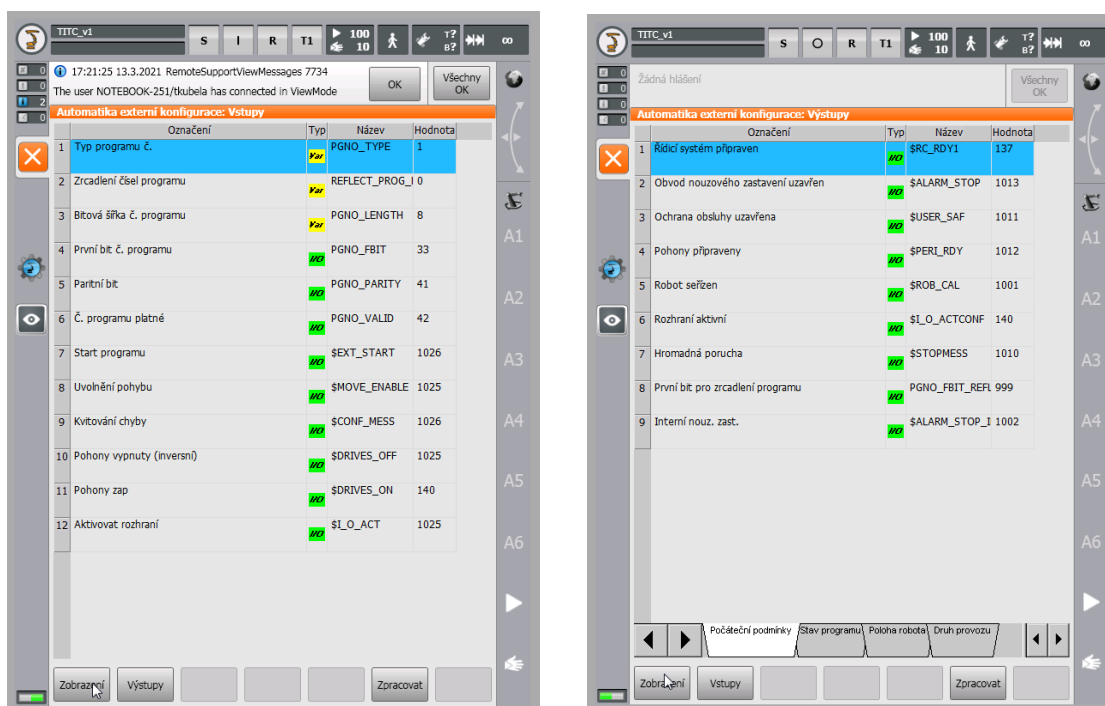
Program WorkVisual bude v tejto práci slúžiť ako prostredie, ktoré komunikuje priamo s robotmi KR5 a KR16-2.

4.2.1 Externá automatika

Pri pohľade na riadiace režimy je možné nájsť nastavovacie režimy T1 – pohyb po jednotlivých bodoch a T2 – pohyb vyššou rýchlosťou po jednotlivých inštrukciách. V tejto práci bude využívaný režim externej automatiky, kde kontrolér robota prijíma povel z nadradeného PLC a posiela ich do príslušného robota, ktorý ich spracuje, odošle spätné hlásky o ich vykonaní spolu s inými signálmi, ktoré kontrolér odošle zase späť do PLC. Signály, ktoré priamo kontrolér robota môžu byť povel na zapnutie či pozastavenie programu. PLC prijíma na druhú stranu signály o stave robota, jeho stavoch v programe alebo iné chybové hlásenia. Volanie jednotlivých programov má za úlohu program CELL.SRC, ktorý bude vysvetlený v ďalšej podkapitole.

4.2.2 Program CELL.SRC

Program CELL.SRC, obr. 37), je určený na výber a volanie programov jednotlivých operácií ako bolo spomenuté vyššie. Na začiatku programu sú inicializačné inštrukcie po ktorých nasleduje opakovacia smyčka. Smyčka začína inštrukciou „LOOP“ a nasleduje požiadavka pre PLC, aby odoslalo číslo príslušného programu. V tomto prípade je to číslo jednotlivých CASE, kde ma každý priradený určitý program ktorý sa po ich vyvolaní začne vykonávať. Tu je potrebné nakonfigurovať nastavenia vstupov a výstupov režimu externej automatiky ako je na obr. 36 a), b).



Obr. 36) a) Nastavenie vstupov externej automatiky, b) Nastavenie výstupov externej automatiky

```

1  &ACCESS RVP
2  &REL 14
3  &COMMENT HANDLER on external automatic
4  DEF CELL ( )
5      EXT Empty()
6      EXT Manipulacia1()
7      EXT Manipulacia4()
8
9  + INIT
13 + BASISTECH INI
18 + CHECK HOME
24 + PTP HOME VEL= 100 % DEFAULT
33 + AUTOEXT INI
36  LOOP
37      P00 (#EXT_PGNO,#PGNO_GET,DMY[],0 )
38      SWITCH PGNO ; Select with Programnumber
39
40      CASE 0
41          P00 (#EXT_PGNO,#PGNO_ACKN,DMY[],0 ) ; Reset Progr.No.-Request
42          Empty () ; Call User-Program
43
44      CASE 1
45          P00 (#EXT_PGNO,#PGNO_ACKN,DMY[],0 ) ; Reset Progr.No.-Request
46          Manipulacia1() ; Call User-Program
47
48      CASE 4
49          P00 (#EXT_PGNO,#PGNO_ACKN,DMY[],0 ) ; Reset Progr.No.-Request
50          Manipulacia4() ; Call User-Program
51
52      DEFAULT
53          P00 (#EXT_PGNO,#PGNO_FAULT,DMY[],0 )
54      ENDSWITCH
55  ENDLOOP
56  END

```

Obr. 37) Program CELL.SRC pre robot KR5 ARC

4.2.3 Vstupné a výstupné signály robotov a PLC

Stavový automat v programe MAIN je zložený hlavne zo signálov, s ktorými PLC komunikuje s kontrolérmi robotov KR5 a KR16-2. Pre správne pochopenie sledu signálov odosielenia a čítania signálov pri externej automatike je nutné si tieto signály bližšie popísať.

Medzi vstupné signály z PLC do kontroléru (KRC) patria nasledovné signály:

PGNO – Bytový signál ktorý uchováva číslo programu.

PGNO_TYPE – Premenná ktorá hovorí, ako sa bude číslo z PGNO čítať. Hodnota 1 je čítanie hodnoty PGNO ako binárne číslo, tzn. PGNO 0001 0000 nesie číslo programu v dekadickom tvare o hodnote 16.

Hodnota 2 je čítanie ako BCD číslo, tzn. PGNO 0001 0000 nesie číslo programu v dekadickom tvare o hodnote 10.

Hodnota 3 je čítanie ako poradie bitu, PGNO 0001 0000 nesie číslo programu v dekadickom tvare o hodnote 5. V tejto práci bude nastavený PGNO_TYPE na hodnotu 3. To bude znamenať že premenné *PGNO_REQ*, *PGNO_PARITY* a *PGNO_VALID* nebudú evaluované a program bude začínať na povel *EXT_START*

PGNO_VALID – Je to vstupný signál, ktorý prenáša kedy má byť čítané číslo programu. Negatívna hodnota tohto signálu znamená že číslo programu je odoslané so zostupnou hranou. Pozitívna hodnota znamená že signál je prenesený s nábežnou hranou. Hodnota 0 hovorí, že číslo je odoslané na nábežnej hrane signálu *EXT_START* (tento signál bude opísaný neskôr).

EXT_START – Je to signál, ktorý spúšťa externú automatiku, v prípade, že je komunikačné rozhranie aktívne (*IO_ACT_CONF* má hodnotu 1).

MOVE_ENABLE – Tento signál je používaný ako povolenie pohybu s robotom. Hodnota *TRUE* dovoľuje pohyb robotom, či už v manuálnom režime alebo v režime externej automatiky. Hodnota *FALSE* hovorí že všetky pohybové jednotky(motory) sú zablokované.

CONF_MESS – Toto je jeden z najdôležitejších signálov, pretože je to signál na kvitovanie chybových hlások. Ak má hodnotu *TRUE*, umožňuje napríklad pohyb s robotom, hneď ako je príčina vzniku chyby vyriešená a eliminovaná.

DRIVES_ON – Zapínanie pohonov, ak je tento signál aspoň 20ms v hodnote *TRUE*.

DRIVES_OFF – Má inverznú funkciu ako vyššie spomínaný *DRIVES_ON* a teda pri aspoň 20 ms trvajúcim pulze sú pohony vypnuté.

IO_ACT – Ak je tento signál v hodnote *TRUE*, je to povel pre aktiváciu automatického externého rozhrania.

Nasledovať budú výstupné signály z kontroléru (KRC) do PLC:

RC_RDY – Keď je signál v hodnote *TRUE*, robot je pripravený na štart programu.

ALARM_STOP – V prevádzkovom stave je v hodnote *TRUE*. V prípade EStopu alebo externého EStopu je tento signál resetovaný do hodnoty *FALSE*.

USER_SAF - V prevádzkovom stave je v hodnote *TRUE*. V prípade porušenia obvodu ochranného oplotenia je tento signál resetovaný.

PERI_RDY – Ak je tento výstup v hodnote *TRUE*, značí to stav, že všetky pohony napájané sú pripravené.

ROB_CAL – V prevádzkovom stave je v hodnote *TRUE*. V prípade že osi robota sú neskalirované, tento signál je resetovaný.

I_O_ACTCONF – Signál, ktorý potvrdzuje aktívnu externú automatiku a signál *IO_ACT* je v hodnote *TRUE*.

STOPMESS – Ak je sa objaví nejaká chyba, napr. EStop či zhodenie ochrany obsluhy, kontrolér nastavuje tento signál na hodnotu *TRUE*. Tento signál sa resetuje po odstránení chyby a nastavením signálu *CONF_MESS* na hodnotu *TRUE*.

ALARM_STOP_INTERN – Signál podobný ako *ALARM_STOP* avšak tento signál je zhodený len pri spustení EStopu. Externý EStop nemá dosah na tento signál.

PRO_ACT – výstup ktorý má hodnotu *TRUE*, hneď ako je aktívny program robota.

PGNO_REQ – Signál ktorý symbolizuje požiadavku robotu pre PLC, aby PLC poslalo číslo programu *PGNO*.

APPL_RUN – Ak je tento výstup aktívny, kontrolér posielá PLC správu, že program v robote bol spustený a beží.

PRO_MOVE – Tento signál má hodnotu *TRUE*, pokiaľ sa robot pohybuje či už v automatickom režime alebo manuálnom.

IN_HOME – Robot sa nachádza v Home pozícií, pokiaľ má tento výstup hodnotu *TRUE*.

ON_PATH – Tento signál je na hodnote *TRUE*, od doby čo bol programový pohyb spustený až po ukončenie programu alebo po reset programu.

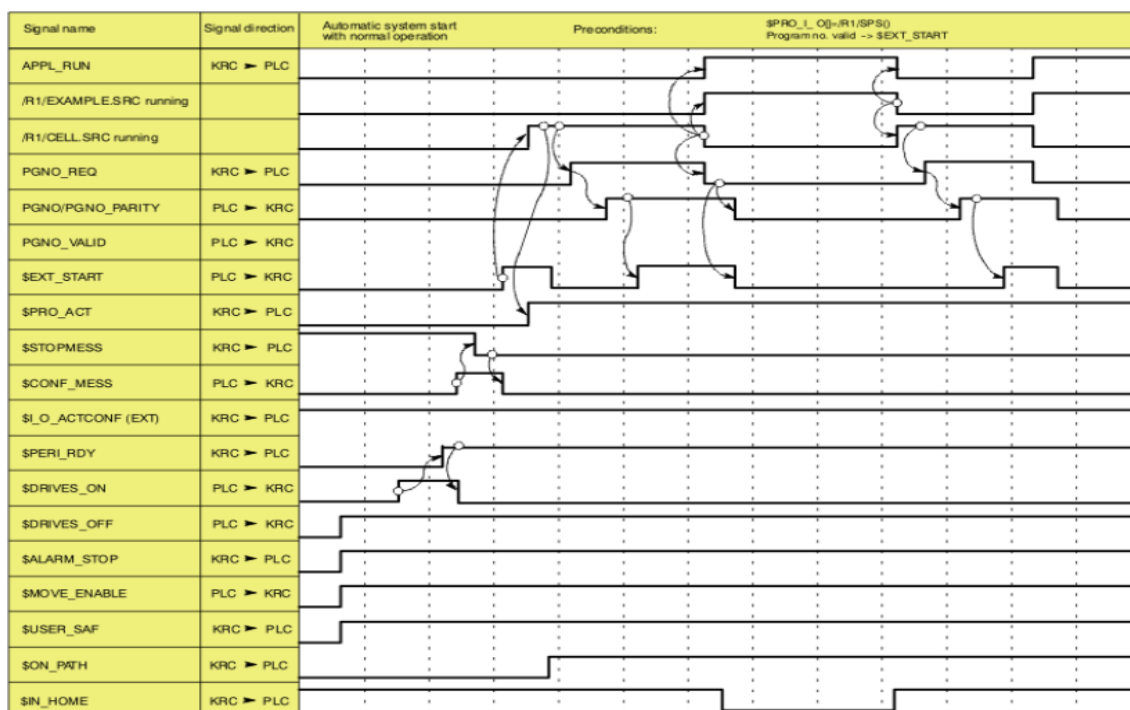
NEAR_POSRET – Keď je nastavená kolízna zóna robota v prípade použitia napríklad viacerých robotov, tento signál má hodnotu *TRUE*, keď sa bude robot nachádzať v tejto zóne. Keď robot opustí túto zónu signál je resetovaný.

ROB_STOPPED – Tento signál je inverzný signálu *PRO_MOVE* a teda hovorí, že robot je zastavený.

EXT – Signál značí že je vybraný mód externej automatiky.

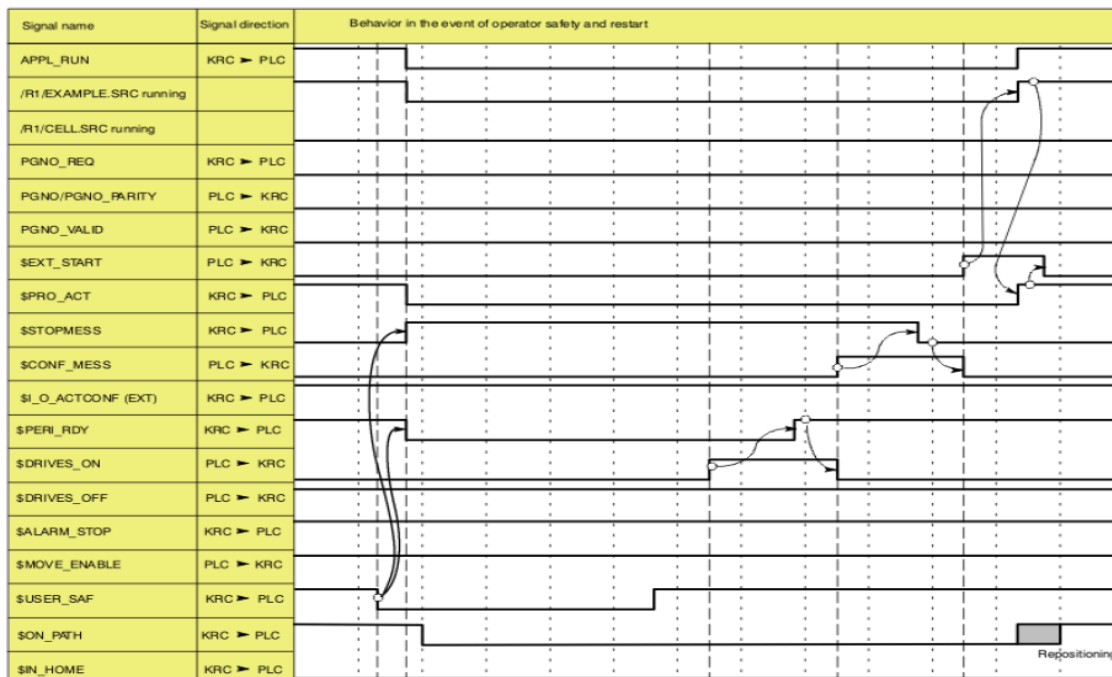
4.2.4 Postupnosť signálov

Externá automatika sa napriek povelom od nadradeného PLC riadi určitou pevne danou logickou postupnosťou signálov. Tieto logiky sa menia v závislosti od podmienok a aktuálnych stavov. Tzn. či bol robot prerušený v priebehu vykonávania programu alebo bol vo vyčkávacom režime, alebo či bolo zastavenie vyvolané prerušením bezpečnosti alebo stlačením tlačidla stop. Obr. 38) a obr. 39) zobrazujú postupnosti signálov robotov KUKA v režime externej automatiky. Obr. 38) pojednáva o správaní signálov pri štarte nového programu, kde na začiatku je vydaný signál z PLC na zapnutie pohonov (*\$DRIVES_ON*). Tento signál aktivuje pohony a z robota do PLC sa odošle signál, že pohony sú pripravené (*\$PERI_RDY*). Nasleduje kvitovanie signálu hromadnej poruchy (*\$STOPMESS*) signálom z PLC (*\$CONF_MESS*). Ak sú pohony pripravené, je možné spustiť program pomocou signálu z PLC (*\$EXT_START*), na ktorý reaguje spustenie *CELL.SRC* programu (*\$PRO_ACT*). Táto sekvencia povelov vyvolá požiadavku na číslo programu od nadriadeného PLC (*PGNO_REQ*). Potom sa spustí vybraný program podľa čísla (*PGNO*) a signál pre bežiacu aplikáciu (*APPL_RUN*) sa nastaví na hodnotu *TRUE* (1 / *HIGH*) a program *CELL.SRC* sa stane neaktívnym (*FALSE* / 0 / *LOW*). Po dokončení aplikačného programu (*APPL_RUN*) padne tento signál do hodnoty *FALSE* a celá sekvencia sa opakuje od zaktívnenia programu *CELL.SRC*, požiadavky čísla atď.. Na obr. 38) je znázornená postupnosť signálov pri aktivácii programu.



Obr. 38) Priebehy signálov pri štarte programu [8]

V prípade, že sa program preruší dynamickým zastavením (Prerušená bezpečnosť obsluhy alebo reštart), sa program obnovuje po pominutí vyvolávajúceho stavu zapnutím pohonov (DRIVES_ON), kvitovania chýb (\$CONF_MESS) a spustenia programu externej automatiky (\$EXT_START). Potom sa stane aplikačný program znovu aktívny. Celá situácia je znázornená na obr. 39).



Obr. 39) Priebehy signálov pri reštarte programu po dynamickom zastavení (Ochrana obsluhy a reštart) [8]

4.2.5 Program .dat

Program vo WorkVisual pre roboty KUKA sa skladá z dvoch základných častí a to zo súboru s príponou .dat a súboru s príponou .src. „Dat“ súbor uchováva všetky informácie o konfigurácii os na dosiahnutie polohy jednotlivých bodov ako je zobrazené na obr. 40). Taktiež tento súbor uchováva dáta o tom ktorý nástroj je používaný, ku ktorým osiam sa vzťahujú jednotlivé body dráhy. Je možné do tohto súboru body vkladať manuálne vpisovaním ich jednotlivých pozícií ale prevažne sa sem zapisujú automaticky vytváraním dráhy robota po bodov v programe s príponou src.

```

69: DECL FDATA TABLE2={TOOL_NO 1,BASE_NO 0,IPO_FRAME #BASE,POINT2[] " ",TQ_STATE FALSE}
70: DECL E6POS XB_TABLE2={X 891.336487,Y 920.204102,Z 293.363770,A -36.0936241,B 30.9893055,C -144.872
71: DECL FDATA FB_TABLE2={TOOL_NO 1,BASE_NO 0,IPO_FRAME #BASE,POINT2[] " ",TQ_STATE FALSE}
72: DECL E6POS XA_TABLE2={X 891.937317,Y 942.288513,Z 271.238190,A -35.6477394,B 31.5340786,C -144.906
73: DECL FDATA FA_TABLE2={TOOL_NO 1,BASE_NO 0,IPO_FRAME #BASE,POINT2[] " ",TQ_STATE FALSE}
74: DECL E6POS XMEDZI_T1T2={X 750.071472,Y 182.897766,Z 513.248474,A -49.5421486,B 1.13261378,C -172.1
75: DECL FDATA FMEDZI_T1T2={TOOL_NO 1,BASE_NO 0,IPO_FRAME #BASE,POINT2[] " ",TQ_STATE FALSE}
76: DECL E6POS XB_TABLE1={X 713.802,Y -791.015381,Z 175.262466,A -90.7735,B 0.645571530,C 178.436279,S
77: DECL FDATA FB_TABLE1={TOOL_NO 1,BASE_NO 0,IPO_FRAME #BASE,POINT2[] " ",TQ_STATE FALSE}
78: DECL E6POS XTABLE1={X 713.512695,Y -789.776489,Z 101.992294,A -90.8199921,B 1.15934336,C 179.07522
79: DECL FDATA FTABLE1={TOOL_NO 1,BASE_NO 0,IPO_FRAME #BASE,POINT2[] " ",TQ_STATE FALSE}
80: DECL E6POS XA_TABLE1={X 714.433105,Y -790.973755,Z 339.460602,A -90.8204651,B 1.21425319,C 179.025
81: DECL FDATA FA_TABLE1={TOOL_NO 1,BASE_NO 0,IPO_FRAME #BASE,POINT2[] " ",TQ_STATE FALSE}
82: DECL E6POS XB_CONV_START={X 1225.27197,Y -103.722809,Z 278.945740,A -90.8204651,B 1.21425307,C 179
83: DECL FDATA FB_CONV_START={TOOL_NO 1,BASE_NO 0,IPO_FRAME #BASE,POINT2[] " ",TQ_STATE FALSE}

```

Obr. 40) Program dat.

4.2.6 Programy .src

Do programu src. sa zapisujú jednotlivé inštrukcie vrátane druhu pohybu (LIN / PTP), názvu dotyčného bodu, rýchlosti robota, informácie o použítom nástroji a iné. Dáta sa môžu zapisovať manuálne pomocou počítača alebo vkladáním inštrukcií priamo z touchpadu robota. Ukážka z programu na ovládanie manipulácie č.1 pre robot KR5 je na obr. 41).

```

1  &ACCESS RVP
2  &REL 121
3  &PARAM EDITMASK = *
4  &PARAM TEMPLATE = C:\KRC\Roboter\Template\vorgabe
5  &PARAM DISKPATH = KRC:\R1\Program\DP_MBONTAR
6  DEF Manipulacia1( )
7  INT
18 PTP HOME CONT VEL=100 % PDAT18
26 ;FOLD % ENTRY TO KOLISION ZONE
27   $out[133]=TRUE
28 ;ENDFOLD
29 PTP B TABLE1 CONT VEL=100 % PDAT8 TOOL[1]:VAKUUM GRIPPER BASE[0]
36 LIN TABLE1 CONT VEL=0.2 M/S CPDAT4 TOOL[1]:VAKUUM GRIPPER BASE[0]
43 ;FOLD % GRIP
44   ;$out[1]=true
45   ;wait sec 0.2
46   ;$out[1]=false
47 ;ENDFOLD
48 ;FOLD % TAKE FROM TABLE1
49   $out[136]=TRUE
50   wait sec 0.7
51   $out[136]=FALSE
52 ;ENDFOLD
53 PTP P51 CONT VEL=100 % PDAT17 TOOL[1]:VAKUUM GRIPPER BASE[0]
60 PTP B CONV START CONT VEL=100 % PDAT11 TOOL[1]:VAKUUM GRIPPER BASE[0]
67 LIN CONV START CONT VEL=0.2 M/S CPDAT5 TOOL[1]:VAKUUM GRIPPER BASE[0]
74 % UNGRIP
79 % PUT TO CONVEYOR START
84 PTP A CONV START CONT VEL=100 % PDAT12 TOOL[1]:VAKUUM GRIPPER BASE[0]
91 PTP HOME CONT VEL=100 % PDAT13
99 % LEAVE KOLISION ZONE
102 END

```

Obr. 41) Program src. pre manipuláciu č. 1 , Robot KR5 ARC

5 VYTVORENIE HMI NA RIADENIE ROBOTICKEJ BUNKY

Na vytvorenie vizualizácie je možné použiť viaceré platformy, či už od výrobcov tretích strán, alebo od firmy Beckhoff ako napr. HMI TwinCAT alebo vizualizáciu priamo v TwinCATE. V tejto práci je použitá posledná spomenutá možnosť navzdory menšej variabilite oproti iným platformám. V tejto kapitole bude vysvetlené vytváranie a funkčnosť vizualizácie.

5.1 Vytvorenie vizualizácie v programe TwinCAT

Všetky dynamické prvky vo vizualizácii musia byť pre správne fungovanie nalinkované na premenné. Je možné prvky nalinkovať priamo na premenné, ale v tom prípade je možné ovládať daný prvok len jednou premennou. V tejto práci je vytvorený špeciálny GVL, ktorý obsahuje len premenné do vizualizácie. Tieto premenné sú nastavené vo vlastnostiach daného prvku v kolónke *Variable*. Ak chceme ovládať daný prvok, je nutné zmeniť hodnotu premennej, ktorá je k nemu nalinkovaná. To sa docieľa pomocou ďalšej POU s názvom *HMI*, ktorá je volaná ako *module call* v hlavnom programe *MAIN*. POU *HMI* je rozdelená na dve časti. V prvej časti sa

načítavajú hodnoty z vizualizácie ako napr. hodnoty prepínačov či textových okien, ktoré slúžia ako vstupne signály.

Ak stlačíme napríklad tlačítko Štart, do premennej *gvl_HMI.Main_Switch1* sa zapíše hodnota TRUE. V programe HMI sa následne načíta táto premenná a zapíše sa do príslušnej premennej, v tomto prípade *gvl.bStart*, ktorá je už implementovaná v PLC programe.

V druhej časti sa zapisujú hodnoty premenných s príponou *gvl*. z PLC programu do vizualizačných premenných s príponou *gvl_HMI*. Na obr. 42) je ukážka z programu HMI, konkrétne zo zapínania kontroliek v sekcii, ktorá ukazuje prebiehajúcu manipuláciu. V tomto programe je použitý aj vytvorený funkčný blok s názvom *Blik*, ktorý po privedení vstupného signálu s hodnotou *TRUE*, striedavo v určitom intervale zapína a vypína výstupnú premennú. Časový interval je možné si zmeniť zápisom do premennej *tPeriod*. Výsledkom tohto funkčného bloku je blikanie kontrolky s určitou frekvenciou. Ak chceme kontrolku ovládať rôznymi signálmi stačí ich napísať do série so spojovníkom AND. V prípade že chceme aby mala kontrolka viacero svetelných módov, je nutné si vytvoriť ďalšie pomocné premenné (*gvl_HMI.Main_Lamp51a*, *gvl_HMI.Main_Lamp51b*). Prvá spomínaná premenná slúži na trvalé svietenie a druhá na blikanie. Do hlavnej premennej *gvl_HMI.Main_Lamp51* zapisujeme obe tieto pomocné premenné ale s logickým OR. Príklad je viditeľný na obr. 43).

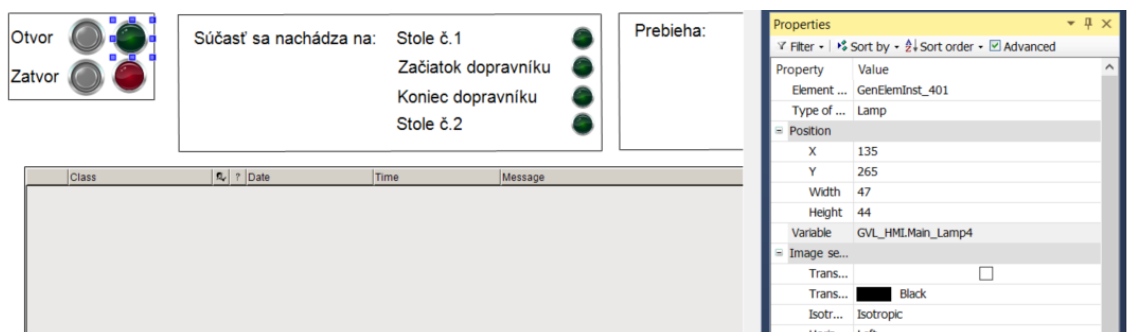
```

3  VAR
4      Blik1: Blik;
5      Blik2: Blik;
6      Blik3: Blik;
7      Blik4: Blik;
8      Blik5: Blik;
9  END_VAR

36 (*-----Prebieha-----*)
37 gvl_HMI.Main_Lamp50 := gvl.bMan_1_Q AND gvl.R1_bAPPL_RUN_I; //trvale svietenie zelenej kontrolky
38 gvl_HMI.Main_Lamp52 := gvl.bMan_2_Q AND gvl.Conveyor_bRun_Q; //trvale svietenie zelenej kontrolky
39 gvl_HMI.Main_Lamp54 := gvl.bMan_3_Q AND gvl.R2_bAPPL_RUN_I; //trvale svietenie zelenej kontrolky
40 gvl_HMI.Main_Lamp56 := gvl.bMan_4_Q AND gvl.R1_bAPPL_RUN_I; //trvale svietenie zelenej kontrolky
41 gvl_HMI.Main_Lamp51a := gvl.bMan_1_Q AND gvl.bStopped; //trvale svietenie žltej kontrolky
42 gvl_HMI.Main_Lamp53a := gvl.bMan_2_Q AND gvl.bStopped; //trvale svietenie žltej kontrolky
43 gvl_HMI.Main_Lamp55a := gvl.bMan_3_Q AND gvl.bStopped; //trvale svietenie žltej kontrolky
44 gvl_HMI.Main_Lamp57a := gvl.bMan_4_Q AND gvl.bStopped; //trvale svietenie žltej kontrolky
45 Blik1(bStartValue:=gvl.bPause AND gvl.bMan_1_Q AND NOT gvl.bStopped, tPeriod:= T#250MS, bCoil:=gvl_HMI.Main_Lamp51b );
46 Blik2(bStartValue:=gvl.bPause AND gvl.bMan_2_Q AND NOT gvl.bStopped, tPeriod:= T#500MS, bCoil:=gvl_HMI.Main_Lamp53b );
47 Blik3(bStartValue:=gvl.bPause AND gvl.bMan_3_Q AND NOT gvl.bStopped, tPeriod:= T#250MS, bCoil:=gvl_HMI.Main_Lamp55b );
48 Blik4(bStartValue:=gvl.bPause AND gvl.bMan_4_Q AND NOT gvl.bStopped, tPeriod:= T#250MS, bCoil:=gvl_HMI.Main_Lamp57b );
49 gvl_HMI.Main_Lamp51 := gvl_HMI.Main_Lamp51a OR gvl_HMI.Main_Lamp51b; //ľtá kontrolka svieti alebo bliká
50 gvl_HMI.Main_Lamp53 := gvl_HMI.Main_Lamp53a OR gvl_HMI.Main_Lamp53b; //ľtá kontrolka svieti alebo bliká
51 gvl_HMI.Main_Lamp55 := gvl_HMI.Main_Lamp55a OR gvl_HMI.Main_Lamp55b; //ľtá kontrolka svieti alebo bliká
52 gvl_HMI.Main_Lamp57 := gvl_HMI.Main_Lamp57a OR gvl_HMI.Main_Lamp57b; //ľtá kontrolka svieti alebo bliká
53
54
55 (*-----R1-----*)
56 gvl_HMI.R1_Lamp1 := NOT gvl.R1_bALARM_STOP_I;
57 gvl_HMI.R1_Lamp2 := gvl.R1_bSTOP_MESS_I;
58 gvl_HMI.R1_Lamp3 := NOT gvl.R1_bALARM_STOP_INTERN_I;

```

Obr. 42) Program HMI na mapovanie signálov z / do PLC do / z vizualizácie

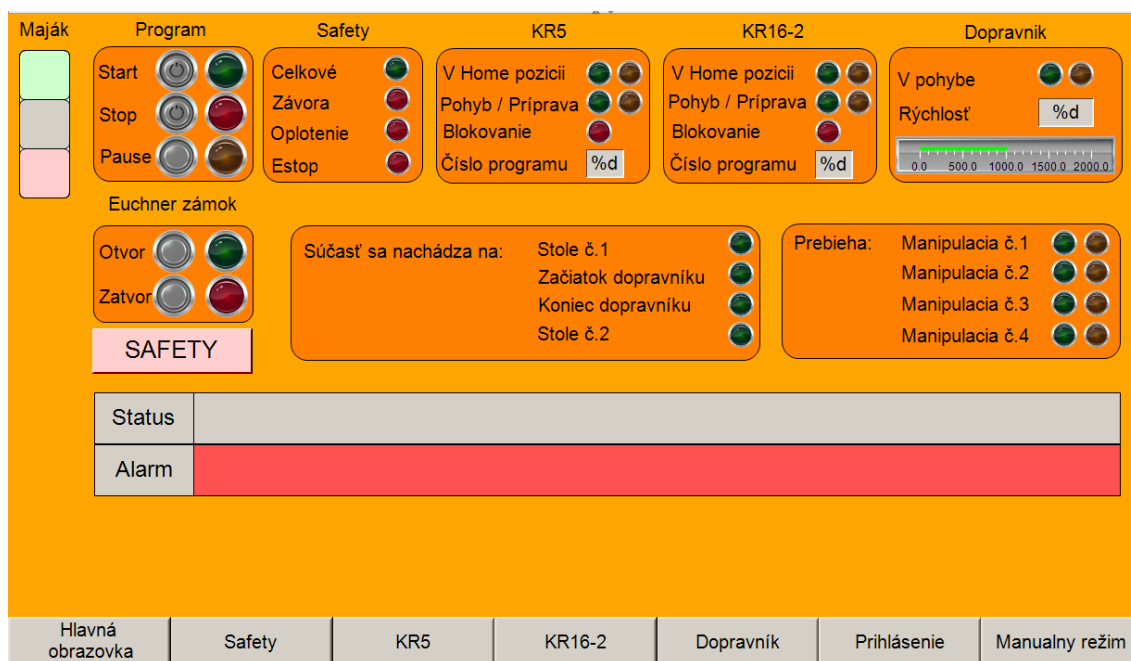


Obr. 43) Nastavovanie premennej vo vizualizácii

5.1.1 Hlavné okno

Hlavné okno, obr. 44), je rozdelené na jednotlivé celky / časti. Na spodnej strane bude navigačná lišta s tlačítkami, ktoré otvoria jednotlivé stránky. V prípade stlačenia tlačítka *Prihlásenie* sa zobrazí dialógové okno na prihlásenie užívateľa do systému. Ak je prihlásený užívateľ, má možnosť posielat' jednotlivé signály do robotov či dopravníku. To je mu umožnené len v manuálnom režime, ktorý je aktivovaný stlačením tlačítka manuálny režim. Kvôli bezpečnosti je toto tlačítko možné stlačiť len ak je prihlásený užívateľ. V momente, keď je aktívne manuálne riadenie, tlačítko *Manuálny režim* svieti na červeno a v oknách KR5, KR16-2 a Dopravník sa zobrazia Checkboxy, ktorými sa nastavujú signály. Nastavovanie signálov funguje tak, že v prípade že je signál nastavený automatikou na hodnotu *TRUE*, po prepnutí do manuálneho režimu bude checkbox na zapínanie signálov na rovnakej hodnote aby sa nenarušila kontinuita programu. Tento signál je potom samozrejme možné zhodiť či nahodiť podľa potreby. Rovnako to platí aj pre signály s hodnotou *FALSE*. Pri deaktivácii manuálneho režimu sa signály riadia opäť podľa automatického programu.

V časti program je jednoduché ovládanie chodu PLC Main programu. Program je možné pozastaviť stlačením prepínaču *Pause*, po jeho opätovnom stlačení program pokračuje v činnosti. Pri stlačení *Stop* je program zastavený a je možné ho znova spustiť až po držaní tlačidla *Štart* na aspoň na sekundy. Pokračovanie programu je možné len v prípade, vyvolávajúce stavy zanikli a nič neblokuje program. Všetky prepínače majú na pravej strane od nich aj kontrolky, ktoré signalizujú stav tlačidiel - stlačené/nestlačené.



Obr. 44) Hlavné okno vizualizácie

Časť Euchner zámok slúži na otváranie dverí bunky. Po stlačení prepínaču *Otvor* sa pošle požiadavka na otvorenie dverí. Dvere sa otvoria kvôli bezpečnosti až keď roboty stoja. Tlačidlo *Zatvor* spôsobí uzamknutie a zaaretovanie dverí aby mohla bunka pokračovať v činnosti, keď sú dvere zatvorené.

Sekcie KR5 a KR16-2 ukazujú stav jednotlivých robotov. Ak svieti zelená kontrolka v prvom riadku, značí to že robot sa nachádza v Home pozícií. V prípade rozsvietenia žltej

kontrolky robot vykonal presun súčiastky a vracia sa do Home pozície. Zelená kontrolka V pohybe znamená, že robot vykonáva programový pohyb a žltá kontrolka signalizuje stav prípravy na spustenie programu, keď robot stojí. Blokovanie vyjadruje znemožnený pohyb z dôvodu EStop, interný EStop či hromadnej poruchy. Pod kontrolkami sa zobrazujú čísla programu, ktorý sa práve vykonáva.

Pri dopravníku sú len dve kontrolky, zelená svieti ak sa dopravník pohybuje, žltá svieti v prípade pozastavenia dopravníku pomocou *Pause*, a v prípade zastavenia z dôvodu Stop alebo porucha začne blikať. Display ukazuje pozíciu súčasti na dopravníku.

Na hl. obrazovke sa taktiež zobrazuje, kde sa nachádza manipulovaná súčiastka a ktorá manipulácia sa práve vykonáva. Ak svieti pri manipulácii zelena kontrolka, manipulácia sa vykonáva bez problémov. Žltá má podobný princíp svietenia a blikania ako žltá kontrolka pri dopravníku. V spodnej časti obrazovky je okno v ktorom sa zobrazujú chybové hlášky a status operácií.

5.1.2 Okno robot

V prípade potreby detailnejšieho poznania celého procesu je možné zobrazit' signály pre roboty KR5, KR16-2, dopravník či safety. Na obr. 45) je zobrazený detailný pohľad do všetkých signálov z robota KR5. Na ľavej strane sú všetky signály, ktoré posiela kontrolér KR5 do PLC a na pravej zase signály, ktoré sa posielajú opačným smerom. Checkboxy sa zobrazujú v prípade, že je nastavený manuálny režim, ktorý bol vysvetlený vyššie.

Robot KR5

Signály z robota

Alarm stop

Hromadná porucha

Interné núdzové zastavenie

Kolízna zóna

Pohony pripravené

Rozhranie aktívne

Robot zkalibrovaný

Program aktívny

Aplikácia práve prebieha

Požiadavna na číslo programu

Pripravený na štart programu

Robot v Home pozícií

Programový pohyb je aktívny

Robot je zastavený

Spustená externá automatika

Spustený mód T1

Spustený mód T2

Spustený automatický mód

Povely do robota

%d Číslo programu

Aktivuj rozhranie externá automatika

Uvolni pohony

Zapni pohony

Kvitovanie chýb

Spusti externú automatiku

Číslo programu je validné

Status

Alarm

Hlavná obrazovka

Safety

KR5

KR16-2

Dopravník

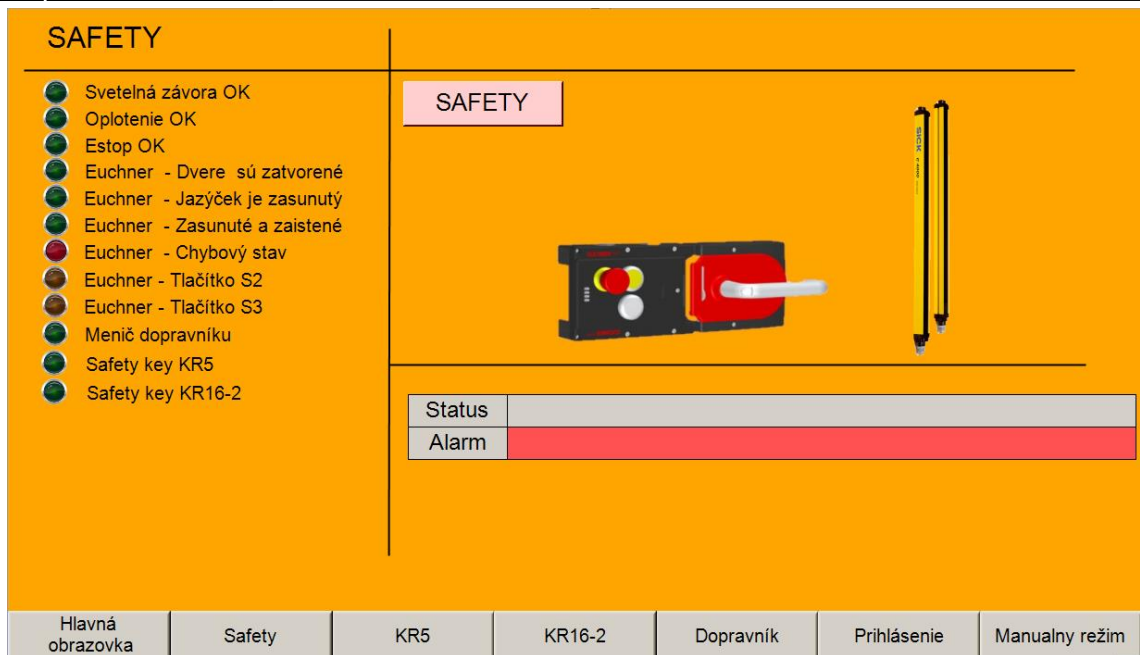
Prihlásenie

Manuálny režim

Obr. 45) Okno Robot KR5

5.1.3 Okno Safety

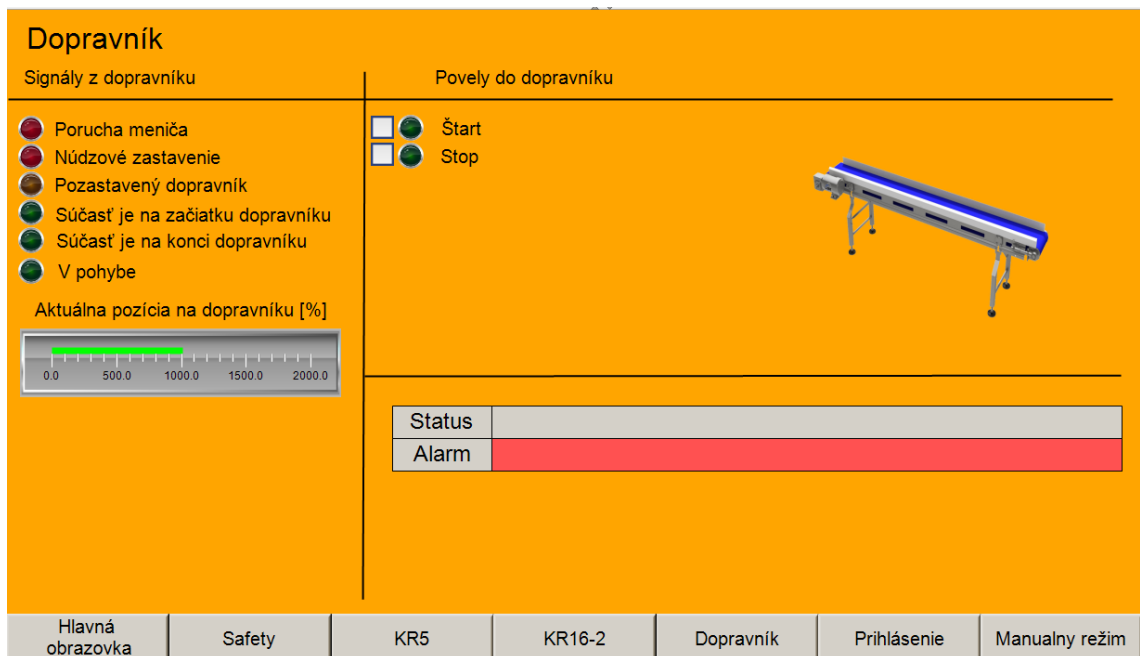
V safety okne sú vytiahnuté všetky bezpečnostné signály. Toto okno je vidieť na obr. 46). Zelené kontrolky svietia v normálnom, bezporuchovom stave a červené v poruchovom stave. Žlte signálové žiarovky symbolizujú tlačítka na elektrickom zámku Euchner.



Obr. 46) Okno Safety

5.1.4 Okno conveyor

Vizualizačné okno dopravník, obr. 47) zobrazuje taktiež signály z dopravníku, a do dopravníku. Taktiež je tu možné navoliť rýchlosť otáčania motora ktorý poháňa dopravník a tým zmeniť rýchlosť pohybu pasu dopravníku.



Obr. 47) Okno Dopravník

5.1.5 Dialógové okno login

Pre odomknutie možnosti priamo aktivovať / deaktivovať jednotlivé výstupy do kontrolérov KRC4 a dopravníku je nutné, aby bol prihlásený užívateľ, ktorý má dostatočné práva na zasahovanie do programu. V praxi mávajú projekty rôzne úrovne od obsluhy , cez

vedúceho smeny až po servis či admina. V tejto práci bude úroveň kvôli jednoduchosti len jedna a to admin. Do tohto účtu je možné sa dostať cez prihlasovací formulár obr. 48), ktorý sa otvorí po stlačení tlačítka *Prihlásenie* v navigačnej lište. Po zadaní údajov sa v prípade zhody nastaví tlačítko *LOGIN* na zelenú farbu ako je na obr. 49). Po stlačení tlačítka LOGOUT sa užívateľ odhlási a padne aj manuálny režim.

Obr. 48) Prihlasovanie okno

Obr. 49) Prihlásený užívateľ admin

Po úspešnom prihlásení je možné navoliť manuálny režim stlačením tlačítka *Manuálny Režim* v navigačnej lište. Po navolení svieti tlačítko červenou farbou ako je na obr. 50). V prípade, že neprihlásený užívateľ sa pokúsi o manuálny režim, toto tlačítko zostane zobrazené šedou farbou a zvolený režim bude naďalej automatický.

Obr. 50) Navolený manuálny režim

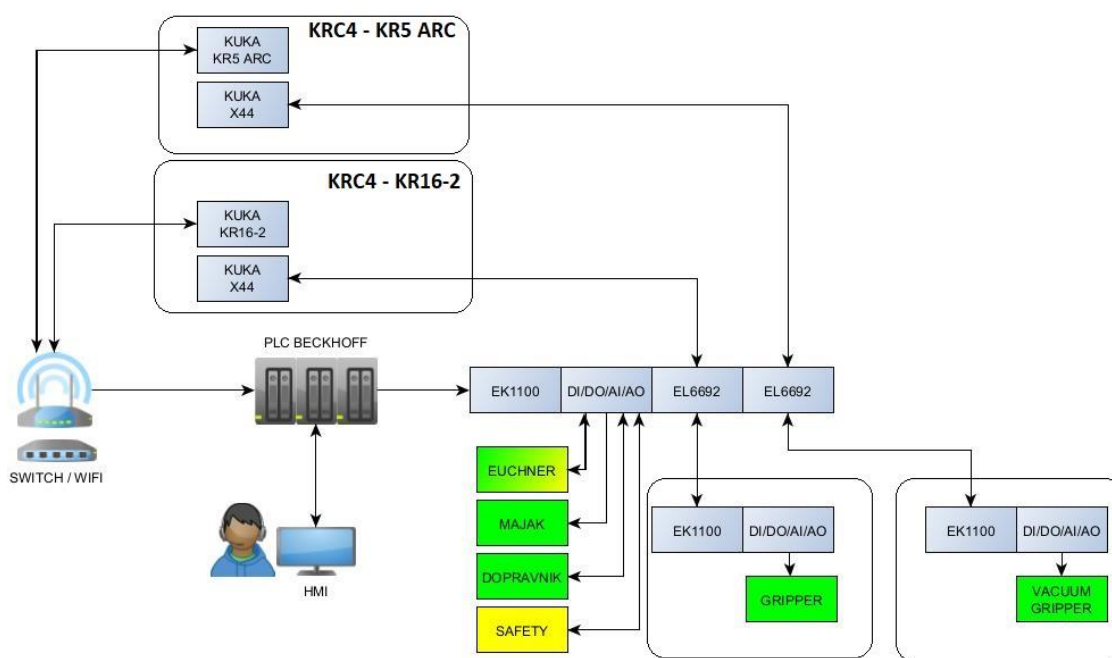
6 IMPLEMENTÁCIA NAVRHNUTÉHO SW

V nasledujúcej kapitole bude riešená implementácia riadiaceho softwaru spolu s užívateľským rozhraním na reálnu robotickú bunku. Na začiatku bude stručný popis potrebných nastavení v TwinCAT, po ktorom bude nasledovať samotné nahrávanie softwaru do PLC a jeho testovanie v praxi.

6.1 Topológia robotickej bunky

V práci je použité PLC od Beckhoffu, ktoré je nadradené nad robotmi a komunikujú spolu cez tzv. „Bridge“ – terminály. V prípade riadiacich systémov robotov ide o terminál s označením EL6692 a zvyšné prvky majú terminál s označením EL1100. Terminály EL6692 umožňujú komunikáciu dvoch EtherCAT mastrov, pretože PLC a aj kontroléry KRC4 robotov KUKA sú mastri. Ak by mala KUKA rozhranie KRC slave, nebol by tento terminál potrebný. Kontroléry robotov majú aj svoje „slave“ a to mosty EL1100, ktoré sú pri päte robotov. S týmito „slave“ si robot posiela signály napríklad pre ovládanie koncových efektorov, v tomto prípade aktiváciu či deaktiváciu vákuových prísaviek.

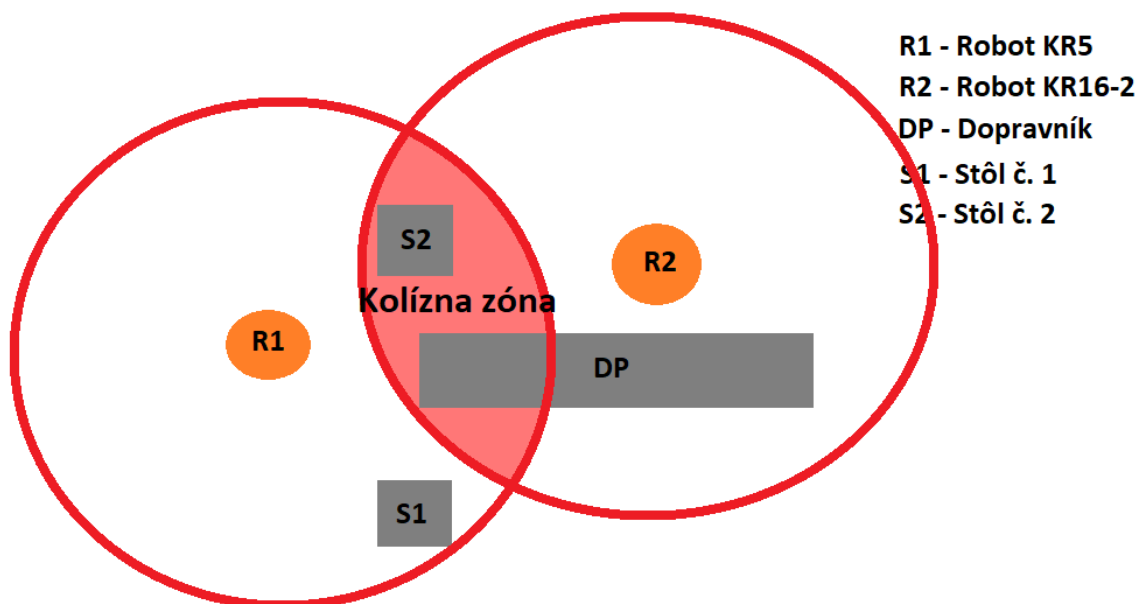
Na terminály sa pripájajú vstupné a výstupné karty. Všetky spojenia prebiehajú cez komunikačné rozhranie EtherCAT. PLC vidí pamäťové rozhranie iba príslušných mostov, kam sú posielané signály z iných rozhraní. Celá topológia robotickej bunky je na obr. 51).



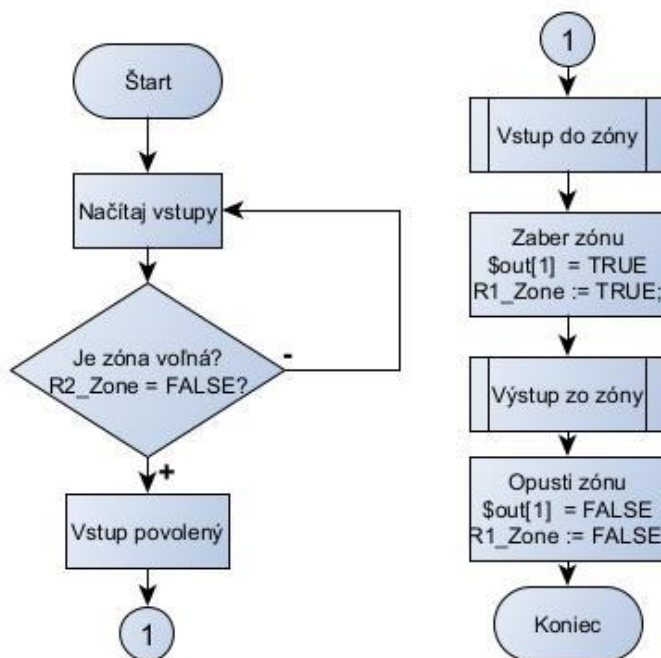
Obr. 51) Topológia robotickej bunky

V prípade, že by si chceli kontroléry robotov prenášať na priamo informácie medzi sebou, bolo by nutné v PLC zaviesť komunikačný modul. Obísť sa to dá samozrejme cez PLC, kde sa pomocou logiky „prepoja“ roboty medzi sebou, kde PLC číta výstupy robota R1, vyhodnotí situáciu a pošle pokyny robotu R2 a naopak. Komunikáciu medzi sebou by roboty potrebovali z dôvodu možnej kolízie, pretože roboty majú v istých polohách rovnaký pracovný priestor.

Z toho dôvodu sú zavedené signály *R1_Zone_Occupied* a *R1_Zone_Occupied*. Tie signály hovoria, či sa roboty v danom čase nachádzajú v kolíznej zóne ako je na obr. 52). Vývojový diagram riadenia robotov z pohľadu voľnosti kolíznej zóny je na obr. 53). V prípade že robot chce začať svoj programový pohyb, musí byť kolízna zóna do ktorej vstupuje voľná. Ak nie je, robot čaká kým bude možné vstúpiť do zóny. Ak je zóna uvoľnená, robot sa rozbehne a začne vykonávať pohyb. V momente ako vstúpi do zóny pošle signál do PLC, že do nej vstúpil a zaberá si ju. Robot zónu uvoľní až tým, že z nej vyjde a signál zhodí do hodnoty *FALSE*.



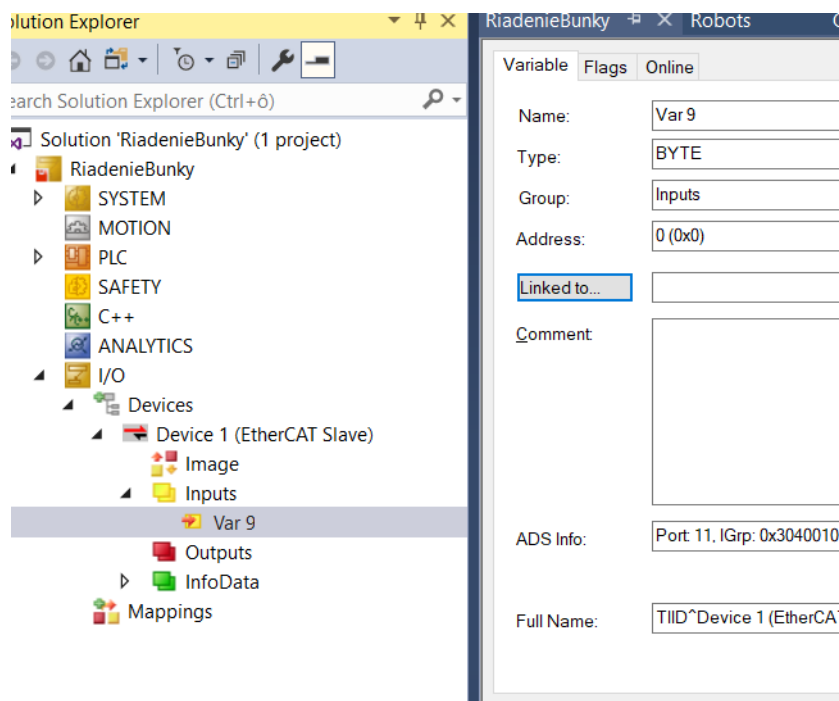
Obr. 52) Kolízna zóna



Obr. 53) Vývojový diagram riadenia robotov v kolíznej zóne

6.2 Mapovanie signálov

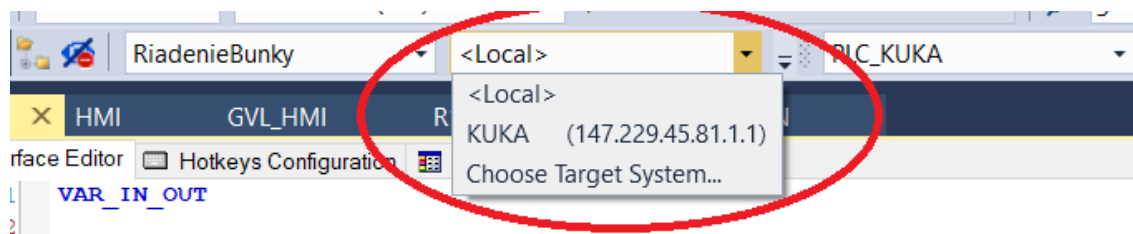
Mapovanie signálov je nutné pre vytvorenie spojenia medzi premennou v PLC a reálnym signálom na I/O termináli. Linkovanie je možné manuálne cez príkaz „Linked to“ ktorý sa objaví po rozkliknutí terminálu až na poslednú úroveň ako na obr. 54).



Obr. 54) Mapovanie premenných

6.3 Nastavenie TwinCATu

Pred nahratím programu do PLC je potrebné vytvoriť cestu, po ktorej sa bude PLC s TwinCATom komunikovať. Po vytvorení cesty sa prepne z „Lokálneho“ PLC (Obr. 55), čo je akoby virtuálne PLC bežiacie v PC na PLC, ktoré je reálne v robotickej bunke. Ak máme aktivovanú konfiguráciu, program skompilovaný, nastavený target KUKA (147.229.45.81.1.1), TwinCAT v Run móde, je možné projekt stiahnuť do PLC ikonkou *LOGIN*.

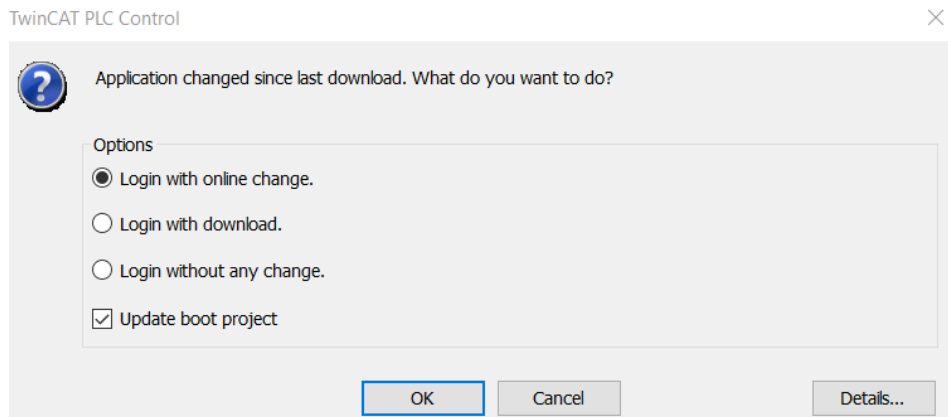


Obr. 55) Nastavenie komunikácie s PLC

Login je možné uskutočniť tromi spôsobmi, ako je na obr. 56):

- Login s online zmenou – Všetky zmeny budú prenesené do PLC
- Login s nahratím – Celý program je prenesený do PLC vrátane inicializácie
- Login bez zmeny – Do Online PLC sa len prihlásime, projekt v PLC ostáva bez zmeny

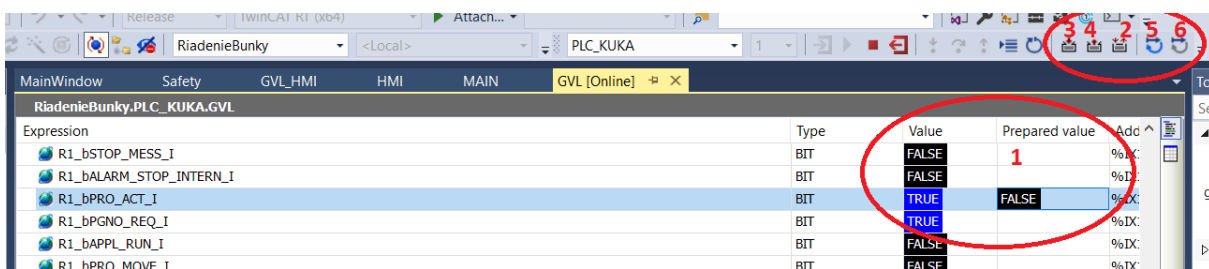
Posledným krokom je spustiť zeleným tlačidlom *START* (zelené „play“ vedľa ikony Login).



Obr. 56) Login do PLC

Po spustení programu sa zobrazia hodnoty premenných, Obr. 57), ktoré je možné editovať vpísaním požadovanej hodnoty do kolónky *Prepared value* (1) a zapísať do PLC pomocou ikony *Write* (2). Ak premennú prepíšeme pomocou ikony *Force* (3), jej hodnota bude trvalá bez ohľadu na požiadavky na prepis od PLC programu. Takúto premennú je v prípade potreby možné odblokovať ikonkou *Unforce* (4). Ak chceme všetky premenné vyresetovať do pôvodnej hodnoty nastavenej na začiatku simulácie, môžeme to uskutočniť tlačítkom *Cold reset* (5), po ktorom je nutné opäť zapnúť simuláciu už len tlačítkom *Start* (Play).

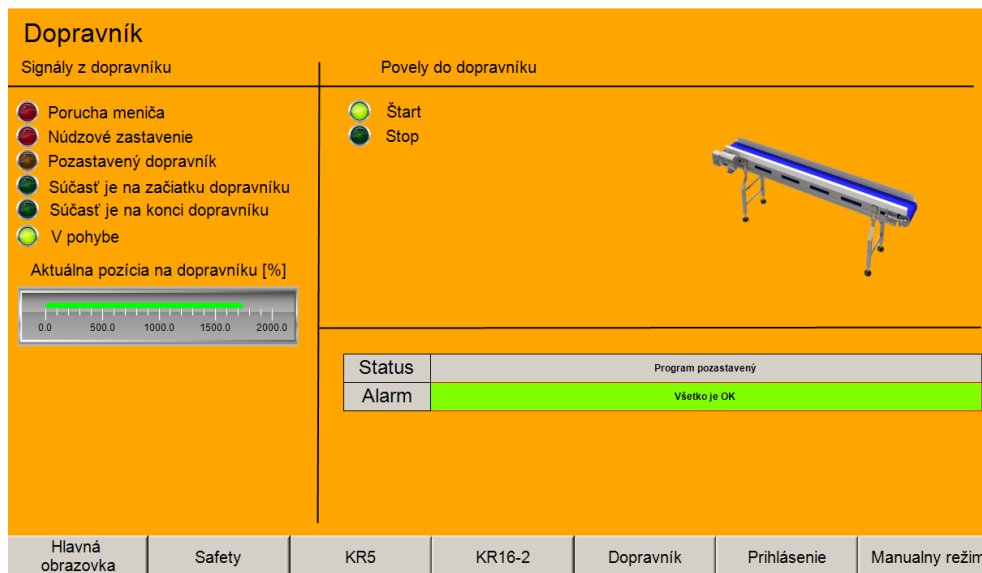
Druhá možnosť je tlačidlo *Reset to origin* (6), ktoré vyresetuje celú simuláciu vrátane vymazania portu 851, takže po jeho použití je nutné nahrať program do PLC pomocou *Login*, vytvoriť port 851 cez vyskočenú hlášku a spustiť tlačítkom *Start* (Play).



Obr. 57) Zapisovanie hodnôt

6.4 Monitorovanie manipulácií

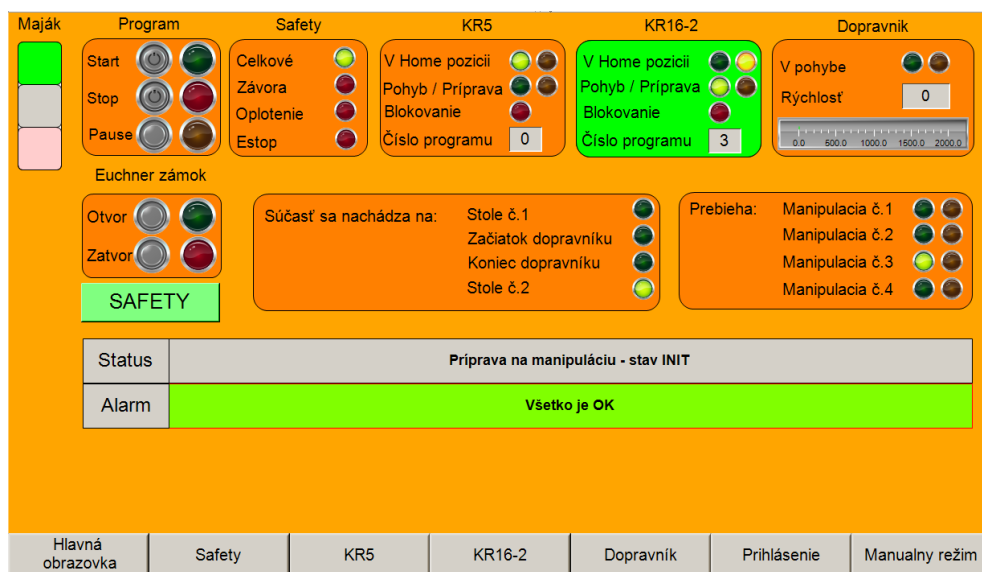
Cely priebeh je jednoducho a intuitívne vidieť z hlavného okna alebo detailnejších okien, ktoré sa aktivujú pomocou tlačidiel v spodnej navigačnej lište. Na obr. 58) je zobrazené okno dopravník, ktorý je v pohybe a presúva súčasť. Aktuálna pozícia je zobrazená displayi.



Obr. 58) Okno dopravník

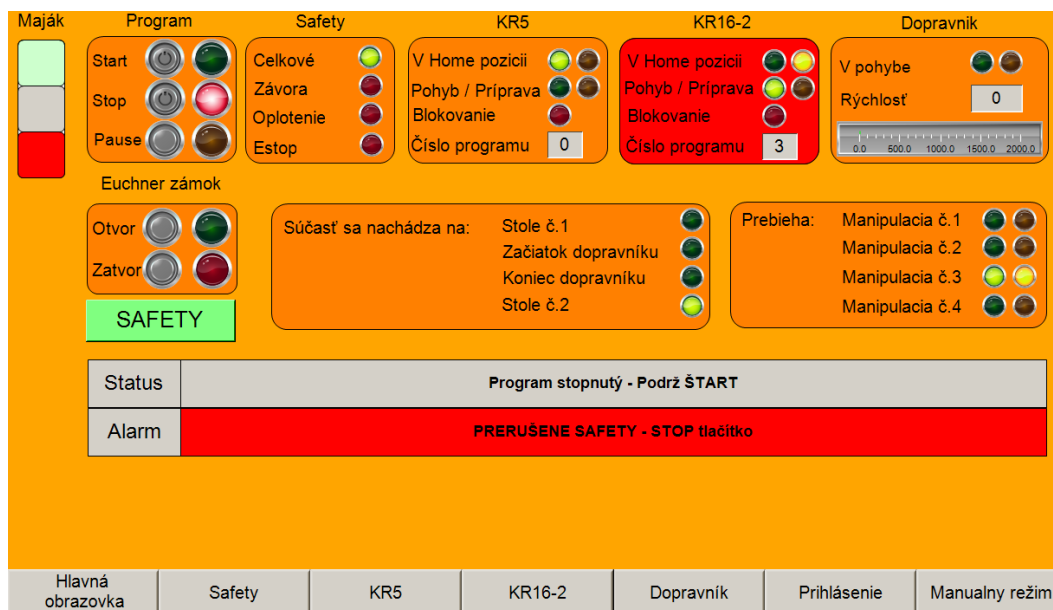
Pre splnenie všetkých safety požiadaviek je potrebné mať zapnuté bezpečnostné tlačítko **SAFETY** či už na hlavnej obrazovke alebo v detailnom okne safety. Ak je toto tlačítko v hornej úvrati, intuitívne svieti zelenou farbou, inak svieti na červeno.

Zelená farba jednotlivých okien signalizuje prebiehajúcu manipuláciu v bezporuchovom stave ako je na obr. 59). V tomto prípade robot KR16-2 položil súčasť na stole č.2 a je na ceste do Home pozície, čo znázorňuje žltá kontrolka. V prípade, že je manipulácia aktívna ale je zastavená alebo pozastavená, okienko svieti červenou farbou ako je na obr. 60).



Obr. 59) Hlavné okno pri manipulácii č.3 robota KR16-2 v bezporuchovom stave

V spodnej časti hlavného okna ako aj v detailných oknách sú zobrazované alarmy a systémové hlášky - statusy, ktoré približujú dianie operácií. Na obr. 59) a 60) je robot KR16-2 pri vykonávaní manipulácie č.3. Na obr. 60) je pri vykonávaní operácie robot stopnutý a na obr. 61) v bezproblémovom chode kde ide pre súčasť, ktorá je na konci dopravníku.



Obr. 60) Hlavné okno pri manipulácii č.3 robota KR16-2 v poruchovom stave



Obr. 61) Robot KR16-2 pri manipulácii s výhľadom na vizualizáciu

7 ZHODNOTENIE A DISKUSIA

Cieľom diplomovej práce bolo navrhnuť a implementovať riadiaci a vizualizačný software v programe TwinCAT na reálnu robotickú bunku. Hlavný doraz pri vypracovaní PLC programu bol na plnú automatizáciu operácií, ktoré vykonáva bunka. Prvým krokom bola dôkladná analýza aktuálneho stavu, v akom sa nachádza bunka aby bolo možné na mieru vytvoriť vhodný program.

Hlavná časť práce bola teda vytvorená v SW TwinCAT od firmy Beckhoff, od ktorej je aj PLC, ktoré riadi činnosť dopravníku a robotov. V tomto SW je vytvorená celá logika postupnosti operácií pomocou programov a funkčných blokov. Čitateľ tu nájde taktiež vizualizáciu slúžiacu ako na monitorovanie prebiehajúcich procesov, tak aj na jednoduché ovládanie formou spustenia či zastavenia jednotlivých manipulácií.

Komunikáciu medzi PLC a kontrolérmi robotov zastrešuje štandardizované spojenie EtherCAT. Programy, ktoré posielajú inštrukcie z kontrolérov do robotov boli vytvárané v programe WorkVisual. V týchto jednoduchých programoch sú deklarované súradnice jednotlivých bodov a postupnosť týchto bodov na dráhe.

PLC program bol nad rámec svojho zadania vytvorený s myšlienkou jednoduchej aplikovateľnosti na iné podobné projekty formou prepísania premenných, ktoré vstupujú do funkčných blokov a vizualizácie a pár ďalších úprav. Nad rámec bolo aj detailnejšie prepracovanie bezpečnostných postupov, ktoré má aj napriek tomu stále dosť priestoru na zlepšenie.

Podľa môjho názoru by sa mohla v rámci iných diplomových, či bakalárskych prác robotická bunka doplniť o snímače prítomnosti manipulovaných dielov rovnako ako aj koncový efektor pre robot KR16-2, vhodný na jednoduché uchopovanie predmetov napr. prísavkami.

8 ZÁVER

Cieľom diplomovej práce bolo navrhnuť riadiaci SW pre robotickú bunku. Následne vytvoriť užívateľské rozhranie a nakoniec obe časti implementovať na reálny HW. Ako uvádza zadanie, SW mal byť vytvorený v programovom prostredí TwinCAT. PLC ktoré bude ovládané je od firmy Beckhoff tak ako aj TwinCAT. Robotická bunka sa skladá z dvoch robotov KUKA a to menovite KR5 ARC a KR16-2, pásového dopravníku, zámku Euchner a ďalších bezpečnostných zariadení ako napríklad svetelná optická závera.

Diplomová práca pojednáva o rešerši z riešenej problematiky, kde sú opísané typy robotov, najznámejší výrobcovia PLC spolu s vývojovými prostrediami pre PLC programy a užívateľské rozhrania. Táto kapitola je ukončená detailnejším popisom HW prvkov, ktoré budú využívané.

Pokračovanie patrí vytváraniu PLC programu, ktorý sa skladá z vytvorených funkčných blokov, podprogramov ako DOOR alebo HMI a programu MAIN, ktorý riadi hlavný stavový automat. V tomto programe sú volané vyššie spomenuté programy a funkčné bloky, čo zabezpečuje kontinuitu programu. Taktiež bol vytvorený program SAFETY, ktorý reprezentuje bezpečnostné prvky, ktoré sú k dispozícii a odráža ich do funkcie celého riadiaceho SW.

Práca sa venuje aj vytváraniu vizualizácie na monitorovanie a čiastočné riadenie chodu bunky. V tejto časti boli vytvorené premenné, ktoré sú zadeklarované v „gvl-liste“ a tie boli nalinkované na jednotlivé prvky vizualizácie ako kontrolky, tlačítka, či textové okná. Vizualizácia sa skladá z hlavnej obrazovky a menších detailných okien, kde sú vidieť všetky prislúchajúce signály.

Ku koncu je opísaný postup implementácie navrhnutého SW pri riadení a monitoringu procesu na reálnu robotickú bunku, ktorá sa nachádza v priestoroch školy.

Na záver je možné potvrdiť, že všetky ciele tejto práce boli splnené a niektoré časti boli spracované do väčších detailov nad rámec zadania, kvôli lepšej interpretácii a predstave o správnom chode robotickej bunky.

9 ZOZNAM POUŽITÝCH ZDROJOV

- [1] TREND, *Pandémia urýchli robotizáciu*. [online], 2020 [cit. 2021-5-6]. Dostupné z: <https://www.trend.sk/spravy/pandemia-urychli-robotizaciju-priemysle-moze-prist-pracutretina-zamestnancov>
- [2] KOLÍBAL, Zdeněk. *Průmyslové roboty I : Konstrukce průmyslových robotů a manipulátorů. 1. vydání*. Brno : VUT Brno, 1993. 189 s. ISBN 80-214-0526-0.
- [3] *Industrial Robots* [online]. ABB, 2021 [cit. 2021-5-7]. Dostupné z: <https://new.abb.com/products/robotics/industrial-robots>
- [4] *Robots* [online]. OH: RobotWorks, 2021 [cit. 2021-5-10]. Dostupné z: <https://www.robots.com/robots>
- [5] *TwinCAT 3 – eXtended Automation Technology (XAT)* [online]. Automa, 2016 [cit. 2021-5-10]. Dostupné z: https://automa.cz/cz/casopis-clanky/twincat-3-extended-automation-technology-xat-2015_06_53801_6979/
- [6] *Siemens TIA Portal – jednotné vývojové prostředí pro automatizaci v průmyslu* [online]. Automa, 2011 [cit. 2021-5-10]. Dostupné z: https://automa.cz/cz/casopis-clanky/siemens-tia-portal-jednotne-vyvojove-prostredi-pro-automatizaci-v-prumyslu-2011_03_43212_6058/
- [7] *Vývojové systémy pro PLC* [online]. Automa, 2011 [cit. 2021-5-10]. Dostupné z: <https://www.automa.cz/soubor-doc716/>
- [8] *KUKA System Software 8.3: Operating and Programing Instuctions for System Integrators*. KUKA Deutschland GmbH, 2018.

10 ZOZNAM SKRATIEK, SYMBOLOV, OBRÁZKOV A TABULIEK

10.1 Zoznam tabuliek

TAB 1) PARAMETRE ROBOTOV KR5 ARC A KR16-2	22
TAB 2) ZÁKLADNÉ PARAMETRE DOPRAVNÍKU DP50	20

10.2 Zoznam obrázkov

OBR. 1) A), B) ROBOTICKÉ PRACOVISKO	19
OBR. 2) PÁSOVÝ DOPRAVNÍK DP50	20
OBR. 3) ELEKTRICKÝ ZÁMOK BUNKY	21
OBR. 4) A) HYBRIDNÝ ROBOT ABB IRB 940 [4], B) PARALELNÝ DELTA ROBOT [3]	21
OBR. 5) A) KUKA KR 5 ARC [4], B) KUKA KR16-2 [4]	22
OBR. 6) UKÁŽKA Z PROGRAMU SIEMENS TIA PORTAL	23
OBR. 7) UKÁŽKA Z PROGRAMU GX WORKS	24
OBR. 8) UKÁŽKA Z PROGRAMU UNITY PRO	25
OBR. 9) UKÁŽKA Z TWINCAT	26
OBR. 10) PREMENNÉ V GVL	27
OBR. 11) VYTŤÁRANIE HMI V TIA PORTAL	28
OBR. 12) VIZUALIZÁCIA V PROGRAME RELIANCE	28
OBR. 13) VIZUALIZÁCIA V PROGRAME VIJEO	28
OBR. 14) A) ST, B) FBD, C) LD	29
OBR. 15) STAVOVÝ DIAGRAM RIADENIA ROBOTOV	30
OBR. 16) ERRORLIST	31
OBR. 17) FUNKČNÝ BLOK BLIK	31
OBR. 18) FUNKČNÝ BLOK ČASOVAČ	31
OBR. 19) VÝVOJOVÝ DIAGRAM PRE FUNKČNÝ BLOK <i>FB_ROBOT</i>	32
OBR. 20) FUNKČNÝ BLOK V PLC PROGRAME	33
OBR. 21) VÝVOJOVÝ DIAGRAM PRE POHYB DOPRAVNÍKA	34
OBR. 22) FUNKČNÝ BLOK <i>FB_CONVEYOR</i>	34
OBR. 23) VÝVOJOVÝ DIAGRAM FUNKČNÉHO BLOKU <i>FB_RECOVERY</i>	35
OBR. 24) FUNKČNÝ BLOK <i>FB_RESET_MAN</i>	35
OBR. 25) PROGRAM DOOR	36
OBR. 26) DIELČIE BEZPEČNOSTI	36

OBR. 27) CELKOVÁ BEZPEČNOSŤ	37
OBR. 28) ENUM E_MANSTATE.....	37
OBR. 29) PROGRAM MAIN	38
OBR. 30) STAV INIT	38
OBR. 31) STAV CONDITION_EVALUATION	38
OBR. 32) STAV NEW_MAN.....	39
OBR. 33) STAV MANIPULÁCIE 1 (MANI).....	40
OBR. 34) STAV MAN2 PRE OVLÁDANIE DOPRAVNÍKA Z PLC	40
OBR. 35) OBNOVA PROGRAMU	41
OBR. 36) A) NASTAVENIE VSTUPOV EXTERNEJ AUTOMATIKY, B) NASTAVENIE VÝSTUPOV EXTERNEJ AUTOMATIKY	42
OBR. 37) PROGRAM CELL.SRC PRE ROBOT KR5 ARC	43
OBR. 38) PRIEBEHY SIGNÁLOV PRI ŠTARTE PROGRAMU [8].....	45
OBR. 39) PRIEBEHY SIGNÁLOV PRI REŠTARTE PROGRAMU PO DYNAMICKOM ZASTAVENÍ (OCHRANA OBSLUHY A REŠTART) [8]	46
OBR. 40) PROGRAM DAT.....	46
OBR. 41) PROGRAM SRC. PRE MANIPULÁCIU Č. 1 , ROBOT KR5 ARC	47
OBR. 42) PROGRAM HMI NA MAPOVANIE SIGNÁLOV Z / DO PLC DO / Z VIZUALIZÁCIE	48
OBR. 43) NASTAVOVANIE PREMENNEJ VO VIZUALIZÁCIÍ	48
OBR. 44) HLAVNÉ OKNO VIZUALIZÁCIE	49
OBR. 45) OKNO ROBOT KR5.....	50
OBR. 46) OKNO SAFETY	51
OBR. 47) OKNO DOPRAVNÍK	51
OBR. 48) PRIHLASOVANIE OKNO	52
OBR. 49) PRIHLÁSENÝ UŽÍVATEĽ ADMIN	52
OBR. 50) NAVOLENÝ MANUÁLNY REŽIM	52
OBR. 51) TOPOLOGIA ROBOTICKEJ BUNKY	53
OBR. 52) KOLÍZNA ZÓNA	54
OBR. 53) VÝVOJOVÝ DIAGRAM RIADENIA ROBOTOV V KOLÍZNE ZÓNE	54
OBR. 54) MAPOVANIE PREMENNÝCH	55
OBR. 55) NASTAVENIE KOMUNIKÁCIE S PLC	55
OBR. 56) LOGIN DO PLC	56
OBR. 57) ZAPISOVANIE HODNÔT	56
OBR. 58) OKNO DOPRAVNÍK	57

OBR. 59)HLAVNÉ OKNO PRI MANIPULÁCIÍ Č.3 ROBOTA KR16-2 V BEZPORUCHOVOM STAVE.....	57
OBR. 60)HLAVNÉ OKNO PRI MANIPULÁCIÍ Č.3 ROBOTA KR16-2 V PORUCHOVOM STAVE.....	58
OBR. 61)ROBOT KR16-2 PRI MANIPULÁCIÍ S VÝHLADOM NA VIZUALIZÁCIU	58

11 ZOZNAM PRÍLOH

Popis prílohy	Názov prílohy
PLC riadenie v programe TwinCAT	RiadenieBunky.tnzip
Program s inštrukciami pre manipuláciu č. 1 vo WorkVisual robot KR5 ARC	Manipulacia1.src
Dátový program pre manipuláciu č. 1 vo WorkVisual robot KR5 ARC	Manipulacia1.dat
Program s inštrukciami pre manipuláciu č.4 vo WorkVisual robot KR5 ARC	Manipulacia4.src
Dátový program pre manipuláciu č. 4 vo WorkVisual robot KR5 ARC	Manipulacia4.dat
Program s inštrukciami pre manipuláciu č. 3 vo WorkVisual robot KR16-2	Manipulacia3.src
Dátový program pre manipuláciu č. 3 vo WorkVisual robot KR16-2	Manipulacia3.dat
Program CELL vo WorkVisual robot KR5 ARC	Cell_KR5.src
Program CELL vo WorkVisual robot KR16-2	Cell_KR16.src
Vývojové diagramy procesov v programe yED	Vyvojove_diagramy.zip

PRÍLOHY